UNIVERSITY OF OSLO
Department of informatics

Relating CORAS diagrams and Markov chains

Master thesis
60 CREDITS

Shahbaz Chaudhary Yaqub
[shahbazy@ifi.uio.no]

1st November 2007

Abstract

"Bismillah ir-Rahman ir-Rahim"
"In the name of Allah, most Gracious, most Compassionate"
-Quran: The Opening (Chapter 1), verse 1

The computer is most certain the best effective tool ever created for mankind. It has brought solutions and shortcuts in order to make life simpler. Because of its enormous concealment computers systems security, meaning the hardware and what is stored inside it, is threatened every second. Our concernment is about how these threats and risks directly affect other computer systems and software. We need to be in control of how to improve the security, and most importantly, how to improve it without making it less user friendly. Before we actually can deal with problems, we have to survey what threats and risks we might face. We have to calculate and evaluate the likelihood of things which might go wrong and how this will have an impact on our system. In order to do all this we have to make use of risk analyses and security seeking techniques.

This thesis presents CORAS analysis, which is a method for risk analyses, and Markov chains, which is a method for probability estimation. The context of the work has been to introduce the idea of an active relation between CORAS diagrams and Markov chains. We have seen through different scenarios if and how they can be related. The result of this thesis is a method including simulation of the system on the basis of CORAS threat diagrams, that produces the needed values to use Markov chains to calculate the probability.

It is surely up to the readers to decide and evaluate the solution I present in this thesis. Hopefully will this be secondary to the contribution which I hope this thesis brings to the research on how to make the probability estimation in CORAS analysis more efficient.

Acknowledgements

First of all I would like to enunciate my gratefulness upon Allah for giving me the strength and encourage to write this master thesis.

I would like to thank my supervisor, Professor Ketil Stølen, for all the great advice and support I got from him. I am very grateful for his feedback and for keeping me focused in my research. He has been a motivator, advisor and a challenger. His interest in my work has really inspired me to work harder and achieve the best of me.

Last but not least, I wish to thank the people who have stood by me everyday. Thanks to my parents, my brothers, my sister, and all my good friends for their outstanding, endless support and encouragement throughout my studies and work with this thesis.

Finally, I owe a lot to three great friends of mine, Wassem Mirza, Omer Siddiq and Anila Khalid. Thank you for your wonderful contribution and support - our conversations, discussions and work together has greatly influenced my work.

Table of contents

LIST OF FIGURES

List of Tables

1. Introduction

When the computer became a common thing in offices and other parts of our lives, we felt that this was the answer to our problems. Internet brought new solutions, information was some clicks away. But these great things brought one big issue to the world: The need of security, and connected to that. How much security do we actually need? We know that too much security can have a negative affect as well. Systems of today are getting more complex each day, and our demands are also growing fast; the system should be more user friendly, and internet connections should work faster and more stable. All this does not make the issue of security easier. When systems become complex it is harder to build a security strategy around it. And our demands force us not to make security too complex: The stricter the security is, the less user friendly will the system may become. For example, will the users have to spend more time on accessing systems, and on the other hand maybe they are not allowed to access some pages on the internet as well.

One thing is for sure, security around systems is of high importance for both companies and private persons. It is necessary for us to have the right knowledge about what the threats could be, and of course how to prevent them. The big question is therefore how the security is maintained; Here it is important to do analysis on how the system works and what the consequences are of the different requirements the system has for the security. Can the system function with a failure in a part of its system? What kind of effect will the failure have on the entire system, and if we in case want to repair the failure, will it then affect the other conditions? If it does, will it have any consequence on the running system? Do we need to shut down the entire system, or could we just terminate some parts of it?

Risk analysis can deliver big benefits to a company. It is a systematic approach to estimate potential losses from threats. Risk analysis can be used for many purposes: In the project stage a company can use it to determine its requirements, or check which risk level is acceptable. Risk analysis can also be used in the operation stage to find out what kind of effect changes will have on the system, or to find the reason for some problems which could occur. In a risk analysis the key elements in each step are confidentiality, integrity and availability. The approach to risk analysis can be divided into two types: Qualitative and quantitative. The difference between qualitative and quantitative analyses is whether we base the analyses on exact numbers or not. [1]

There are many different kinds, and one of them is CORAS [5]. CORAS is a method for security risk analysis. CORAS was built with the aim to be a less time consuming and costly method for risk analysis. Here we examine threats, vulnerability and the control around this. What can go wrong with the system, what part of the system is most vulnerable, and how that can be prevented, and now how we can reduce probability of attacks? CORAS diagrams are an effective way to do risk analyses, it gives a way to calculate and rank the risk and that gives us a figure to follow.

In this thesis we will be looking at Markov chains. Can Markov chains be of any help when calculating the probability in CORAS threat diagrams? What has to be done to utilize Markov chains capability in CORAS threat diagrams?

Markov chains are well known in the mathematic world to do probability equations, and it shows how a deflection in one state can affect the other state, or the whole system. Many analyse methods lead to optimistic predictions for the system, because they assume that the components to be independent. While Markov-analysis look at the reliability and availability of systems, where the components exhibit strong dependence. One basic thing used here is state transition diagram to show different states of the system. This gives an understandable diagram where one can see how the system reacts if a problem occurs in one of the system states.

The thesis is structured in this manner:

Chapter 1 Introduction: *Introduction for the thesis*

Chapter 2 Background: *Background information about CORAS and Markov chains. It will also give an introduction to two other analyze methods: HAZOP and FMEA/FMECA*

Chapter 3 Problem analyses: *Here the problem description is revealed and the purpose of the thesis*

Chapter 4 Relating CORAS diagrams and Markov chains: *This chapter will relate CORAS diagrams with Markov chains. Here we will do a case study and later introduce Markov chains in that case study*

Chapter 5 Method: *We will give formulate a method for how to use Markov chains in CORAS diagrams*

Chapter 6 Implementation, design and tools: *Documentation of the implementation, design and the tool*

Chapter 7 Discussion: *Evaluate experiences and findings during this thesis*

Chapter 8 Conclusion: *What can be done for further work and concluding what the thesis is about*

References

Appendix

User manual

2. Background

“It isn't that they can't see the solution. It is that they can't see the problem." [3].

This is what risk analysis is about. We have to find out what the problem is, where the risks occur, and how to treat them.

Risk analysis starts with a description of the situation which gives us an insight. Risk analysis indicates what the threats could be, the reason for them to occur, the consequences they will have and how they could be prevented. Normally there are five phases in a risk analysis: Context establishment, risk identification, risk estimation, risk evaluation and treatment identification. These five phases are actually enough to determine the result of the risk analysis.

Risk analysis can looked as a sub-activity of risk assessment and risk management. In risk assessment we try to estimate the risk or consider what the risk is. In risk management we try to find out how to prevent the risk. Risk management is built on risk assessment. The five phases mentioned above are a part of both risk assessment and risk management. The first four comes under risk assessment, and the last phase of treatment comes under risk management.

Three questions that should be asked in a risk assessment are: What can go wrong? What is the likelihood that it would go wrong? What are the consequences? [8]
These questions help to identify measure, quantify and evaluate risk and their consequences and impacts.

In risk management three other questions are important: What can be done and what options are available? What are the tradeoffs in terms of costs, benefits, and risks? What are the impacts of current management decisions on the future options? [8]
The last question is important how the treatment suggested by the analyze-team will have an impact in the future.

Further in this thesis we are going to find out how CORAS diagrams gives answers to many of the question asked during the risk analysis. But first I will present different leading risk analysing techniques in order to give a broader picture of how a risk analysis can be done, and how these techniques seek to find out the threats. Do they give us significant answers and if so, then from which angle do they approach the problem.

2.1 Hazard and Operability Analysis

“HAZOP” is short for Hazard and Operability Analyses, and is a technique based on systematic process to identify possible deviations from normal operations and ensure that appropriate safeguards are in place to help prevent accidents. The technique makes use of guide words and is carried out by a team/group during a set of meetings.

The background for HAZOP analyses is the idea of how team approach to an analysis is more efficient then individuals working separately followed by combining their results at the end of the working period. A team for HAZOP analysis includes individuals from different environments and expertise. In order to achieve good and effective teamwork the number of members within a group should be from 4 to 7. The background of these members should consist and vary from project engineer, machinery engineer, instrument engineer to safety engineer. The group should also consist a group leader with some experience with HAZOP.

When you are working with HAZOP analysis, a brainstorming is a useful pre-work activity. The guide words are used in HAZOP analysis to make this brainstorm process more efficient. The guide words are used, in turn, to all the parameters, in order to identify deviations from the design/process purpose. The deviation as mentioned earlier is where the process condition may depart from their design/process objective. The parameters will diversify depending on what type of process is being analysed and process intent.

The guide words used are [9] [10]:

- No – Abnegation of the aim
- More – Quantitative increase in a parameter
- Less – Quantitative decrease in a parameter
- As well as (more than) – An additional activity occurs
- Part of – Qualitative decrease (Only some of the design intention is achieved)
- Reverse – Contradictory of the intention
- Other Than – Absolute substitution (another activity takes place)

Some of the parameters that can be used are [9]:

- Flow
- Temperature
- Pressure
- Phase
- Level
- Relief
- Instrumentation
- Maintenance
- Addition
- Safety
- Reaction

The procedure for using the HAZOP technique can be divided into this [10]:

1. Divide the system into sections (i.e., reactor, storage)
2. Choose a study node (i.e., line, vessel, pump, operating instruction)
3. Describe the design intent
4. Select a process parameter
5. Apply a guide word
6. Determine causes
7. Evaluate consequences/problems
8. Recommend action: What? When? Who?

9. Record information
10. Repeat procedure (from step 2)

The table gives a wider picture of how HAZOP technique has to be applied [11]:

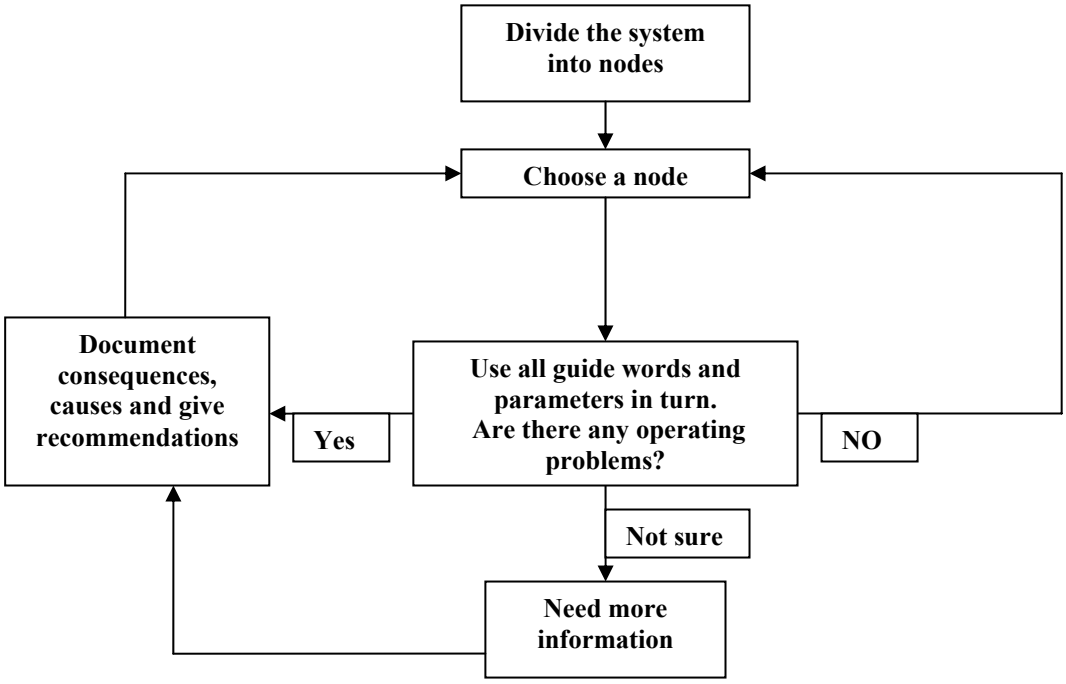


Figure 2-1: Flowchart diagram for the HAZOP technique

Guide words	Deviation	Consequences	Causes	Suggested action
No				
More				
Less				
As well as				
Part of				
Reverse				
Other than				

Table 2-1: Table for HAZOP

HAZOP is a qualitative analysing method which is primarily used for identifying safety hazards and operability problems of process systems, especially in chemical process systems, but it is also used in software systems. HAZOP is time consuming, and it is essential to have access to detailed design and operational information. HAZOP focuses on discovering single failures that could lead to accidents of interest.
For more detailed study of HAZOP see paper [10] and [12].

2.2 Failure Modes and Effects Analysis and Failure Modes, Effects and Critically Analysis

The term “FMEA/FMECA” stands for “Failure Modes and Effects Analysis” and “Failure Modes, Effects and Critically Analyses”. FMEA/FMECA is a method which is one of the most used methods for performing reliability analyses. FMEA is a qualitative analysis while FMECA is an extension of FMEA, and it focuses on the quantitative parameters on the probable failure mode. FMECA adds probability of occurrence and severity of failure to the FMEA process. This analyse method should be implemented at the beginning of the design phase. This way it would give more impact on the final design and identify potential design weaknesses. There is generally a smooth transition between FMEA and FMECA, we will refer to both methods as FMECA further in this paper.

When performing FMECA analysis there are some basic steps which should be followed [13]:

- Assemble the team
- Establish the ground rules
- Gather and review relevant information
- Identify the items or processes to be analysed
- Identify the functions, failures, effects, causes and controls for each item or process to be analysed
- Evaluate the risk associated with the issues identified by the analysis
- Prioritize and assign corrective actions
- Perform corrective actions and re-evaluate risk
- Distribute, review and update the analysis, as appropriate

The purpose of FMECA is to take actions in removing or reducing failures, starting with the highest-priority ones. FMECA uses a worksheet to do the analysis, and before working with this worksheet one should go through some certain steps [11]:

i. The system must be divided into subsystems that can be analysed separately

ii. A function diagram has to be made that shows the dependencies between the subsystems from i), and how they are related together.

iii. A component overview is made for each subsystem

Description of entity			Description of failure			Effect of failure		Failure rate	Failure effect scale	Recommended actions	Remarks
Ref. nr	Function	Process state	Failure mode	Cause of failure	Detection of failure	On other entities	On main function				

Table 2-2: An example a worksheet [11]

Even tough the FMECA can be time consuming; it studies every single failure separately as an independent event. FMECA is incredibly effective when it is used on system where single-component failure is most likely to occur. One of the major drawbacks of FMECA is that all component failures are analysed and documented, and that includes even those failures that have small or nearly no consequences. This creates a large amount of unnecessary documentation if there is a large system that is being analysed.
For more detailed study on FMECA see [11] and [14].

2.3 CORAS analysis

CORAS is a methodology for model-based risk analysis [4]. It is largely based on the acknowledged *Australian/New Zealand Standard 4360:2004* for risk management [17]. The goal of CORAS is to develop a better methodology for more precise and effective risk analyses of security critical IT systems. This will prevent companies to use large sums on security issues, and by using this methodology in an early face they will see what kind of risk exists, and how to deal with them.
The methodology of CORAS can be divided into:

- Identify context: Characterize the target with analyses, what is the focus and the scope of the analyses. What losses can the client tolerate, since it will always be risk involved?

-

- Identify risks: Identify threats to assets with for example brainstorming, and also identify vulnerabilities of the assets.

- Estimate risk level, evaluate risks: All risks can not be eliminated, so we have to decide which risk that needs treatment. We have to know about the risk levels.

- Treat risks: Identify treatment of unwanted risks. Evaluate and prioritise different treatments.

As mentioned in these points, CORAS objects are to develop a structure to make well-organized risk analysis. This framework can be used by any existing company to analyse their risk levels, and how they can be treated. Notifying the results of an analysis in such way that they are well understood by users can be challenging. To simplify this CORAS has a UML based language which targets security risks. The CORAS language offers treatment overview diagrams. Treatment overview diagrams may for example be used to provide a high level summary when presenting the main findings from an analysis.

The CORAS language offers five kinds of diagrams: Asset, threat, risk, and treatment diagrams. The asset diagram contents of what is valuable for the client and needs protection, the assets must be accurate so that the analysis does not fail. In the threat diagram we show something or someone that will do harm to the assets. It also shows where the vulnerabilities could be in the system and how a threat can act. In the risk diagram the consequences are estimated, and it gives an overview of acceptable and non-acceptable risks. The treatment diagram shows where the treatment should be placed to indulgence the risk.

The structure of CORAS can be separated in three different apparatus:

- *Risk modelling language:* both the graphical and the textual syntax of the CORAS diagrams and semantics.

- *Method:* description of the security analysis process, and guidelines for making CORAS diagrams

- *Tool:* for documenting, maintaining and reporting risk analysis results

Working with security analysis can be confusing for many people. Most certainly it is because the importance of finding out the right target for the analysis. One should know where to start the analysis and how to follow through the whole process. This is important to know to actually figuring out the correct answer in the end of the analysis. Many people seem to be failing doing this chore because of lack of experience in this field. They also might start at the wrong end, or they can get confused in the middle of process around different equations.
We will try to give some guidelines of how to use CORAS in a security analyses, this can be divided in seven steps: [5]

1. To get the initial information about what you are going to analyse, you should meet the company you are going to do the analyses for, and let them give you an overall

view of the company and their goals of the analyses they expect. They should also present the target they wish to be analysed.

2. Go through and sort out the information you achieved from the first meeting, and then set up a second meeting where you present your personal understanding of the information. You should make a rough, high-level security analysis. This will clarify any misunderstandings from the first meeting, and help you when making the more detailed analysis later.

3. Here you will present a more distinguished description of the target to be analysed, and all assumptions and other preconditions being made. The client has to approve all this before you can go to step four.

4. Here we put in order a workshop among people with expertise on the target of evaluation. The purpose is to identify as many potential unwanted incidents, threats, vulnerabilities and threat scenarios as possible.

5. Workshop is used in this step also. Now the focus is on estimating consequences and likelihood values for each of the identified unwanted incidents.

6. You will now give the client the first overall risk picture. This is important because it will produce necessary adjustments and corrections.

7. This last step is best done in a workshop. You will go through treatment identification and address cost/benefit issues or the treatments.

There are used different icons in the CORAS diagrams; this gives an abstract view of the risk analysis. The icons are easy to read and the relationship between the icons is straightforward and therefore easy to understand. The different icons which are used in CORAS diagrams are shown in the forthcoming figure.

Treatment scenario: The way to deal with a risk

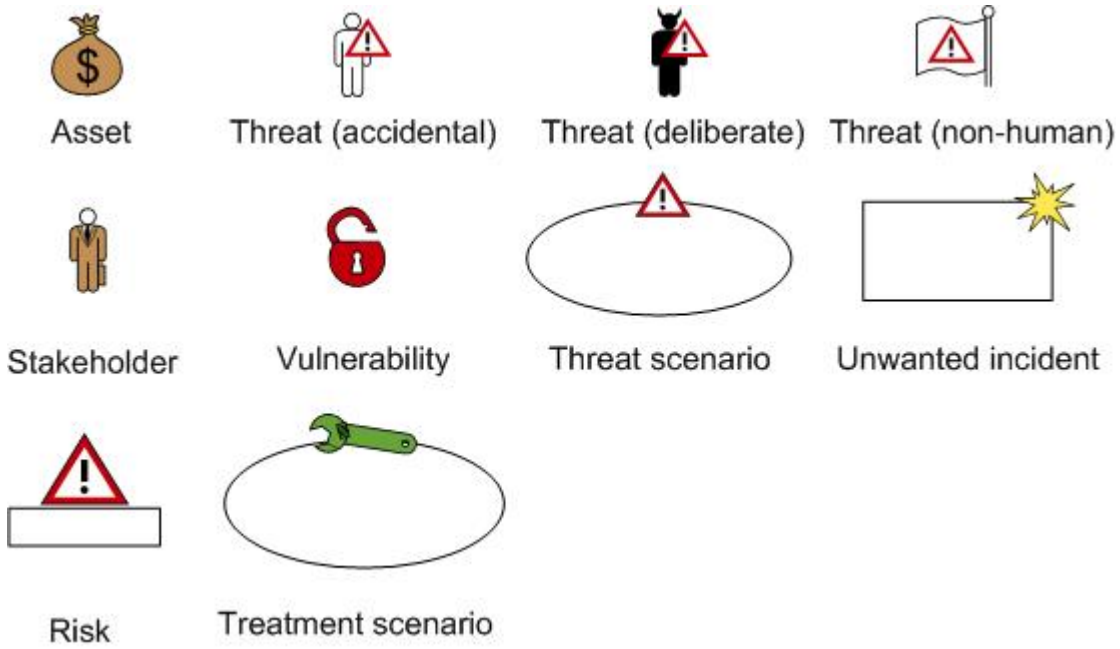


Figure 2-2: CORAS constructs.

- **Asset:** Something that is of value for the stakeholder and requires protection.
- **Threat:** A possible cause of an unwanted incident.
- **Stakeholder:** People or organisation who may be affected, or may influence, an assessment or activity concerning the target of the analysis.
- **Vulnerability:** Weakness which can be exploited by threats
- **Threat scenario:** Something that can occur because of a threat exploiting some vulnerabilities.
- **Unwanted incident:** Something we want to prevent, since it can cause harm to the assets
- **Risk:** The possibility of something happening that will impact the assets.

2.3 Markov analysis

As mentioned in the introduction we will be looking at Markov chains. Andrei Andreevich Markov (1856-1922) was a well known Russian mathematician. Solutions for many problems in today's science and technology would not be possible without his contributions.

In mathematics, a Markov chain is a discrete-time stochastic process with the Markov property. Having the Markov property means the next state solely depends on the present state and doesn't directly depend on the previous states. At each point in time the system may have changed states from the state the system was in the moment before, or the system may have stayed in the same state. The changes of state are called transitions. If a sequence of states has the Markov property, then every future state is conditionally independent of every prior state.

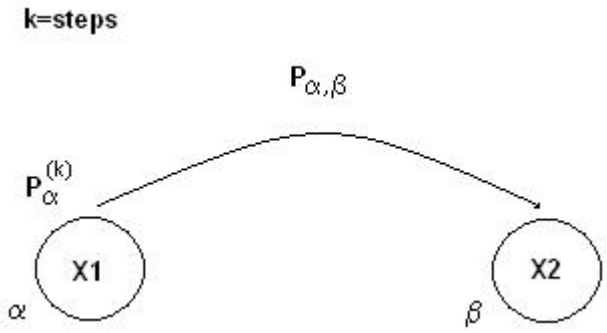
Most of Markov's works were committed to simple homogeneous chains[1], he determined probability determined probability $p_{\alpha,\beta}$ of the event $x_{k+1}=\beta$ given that $x_k = \alpha$, that $p_{\alpha}^{(k)}$ of the events $x_k=\alpha$ are connected by the simple formula

$$p_{\beta}^{(k+1)} = \Sigma p_{\alpha}^{(k)} p_{\alpha,\beta}$$

Explanation for the formula:

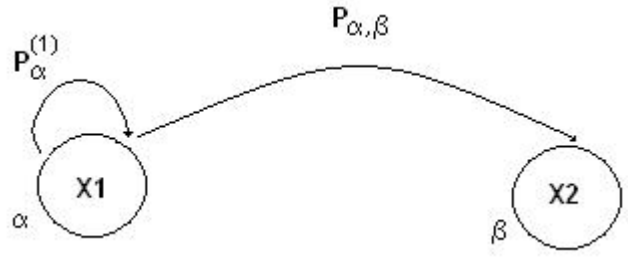
$$p_{\beta}^{(k+1)} = \Sigma p_{\alpha}^{(k)} p_{\alpha,\beta}$$

k = steps that state 1 makes to go to state 2



If let k = 1 we will get:

$$p_{\beta}^{(2)} = \Sigma p_{\alpha}^{(1)} p_{\alpha,\beta}$$



Markov-analysis is something that has developed over many years based on work by Markov on probability. Markov-analysis is a special type of stochastic process[2]; this means that it can determine the future behaviour of a condition by its present state. A stochastic matrix is a

[1] Markov called the chain homogeneous if the conditional distributions of x_{k+1} given x_k were independent of k.

[2] A process where incidents occurs randomly

square matrix whose columns are probability vectors. A Markov chain is sequence of probability vectors $x_0, x_1, x_2, ...,$ together with a stochastic matrix P, such as;
$x_1 = Px_0, x_2 = Px_1, x_3 = Px_2$

We can define a stochastic matrix like this:

A matrix $A = (a_{ij})$ is a stochastic if $a_{ij} \geq 0$ for $i, j = 1,\ldots,n$ and $\Sigma_j a_{ij} = 1$ for $i=1,\ldots,n$

When a Markov chain of vectors in R^n describes a system or a sequence of experiments, the entries in x_k list, are the probabilities that the system is in each of n possible states, or the probabilities that the outcome of the experiment is one of n possible outcomes. [6]

One thing which is important to remember is that previous state does not effect the determination of the current state; we only look at the present state. The Markov method is especially usable for smaller systems with complex maintenance strategies. In larger systems it is difficult to construct diagrams, and that is a major drawback of the Markov method. On larger systems you can use a combination of Markov method and simpler quantitative methods.

In homogeneous Markov models the failure and repair rate of a component can depend on the current state. This can create some limitations because of some assumptions.
Probabilities changing from one state to another are believed to remain constant, and a Markov model is used when a constant failure rate and repair rate hypothesis is justified. Probabilities are determined only by the present state and not by the systems passed history, so the future state of the system is understood to be independent of all but the current state of the system. [2]

As mentioned earlier most of Markov's work was committed to homogeneous chains, but he developed also theory on non-homogeneous chains. The difference between those two is that a homogeneous chains is characterized by constant transition rates between the states, and a non-homogeneous chain is regarded as by the fact transition rates between the states are functions of a global clock e.g., beyond mission time.

Markov chains can be divided into discrete and continuous-time, where discrete can be classified as irreducible or reducible, periodic or aperiodic, recurrent or transient [2]. While continuous-time Markov chain is completely determined by its transition times and the transition probabilities, the state transitions may occur at any time and the time between transitions is exponentially distributed. For some examples of irreducible or reducible, periodic or aperiodic, recurrent or transient, you may see the Appendix A.

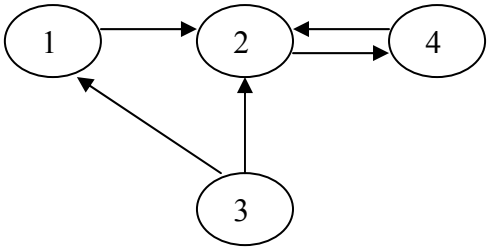
R. A. Howard[3] gives a nice description of Markov chain: As a frog jumping on a set of lily pads. The frog jumps from a lily pad to a lily pad with the accurate transition probability. We can illustrate a Markov chain as this: $S = \{s_1, s_2, \ldots, s_r\}$ is a set of sates. The process starts in one of these states and moves from one state to another, from state s_i to state s_j with a probability p_{ij}. This probability, p_{ij}, called *transition probability*, does not depend upon which states the chain was in before the current state. The process can remain in the state it is in, and this occurs with probability p_{ij}. It should be clear that, when a Markov chain

[3] R. A. Howard, Dynamic Probabilistic System, vol.1, John Wiley and Sons 1971

jumps, the distribution of where it lands depends only on where it was at the time when it jumped and not on where it was in the past.

Transition probability matrix[4] is commonly used when working with Markov chains. This gives the probability that the Markov chain will move to state **j** at time **n+1** given that it is at state **i** at time **n**, independent of where it was prior to time **n**.

Example 2-1: A Markov chain



This is a discrete Markov chain with four states (1, 2, 3, 4), it has the following matrix:

$$P = \begin{pmatrix} 0 & 0 & p_{13} & 0 \\ p_{21} & 0 & p_{23} & p_{24} \\ 0 & 0 & 0 & 0 \\ 0 & p_{42} & 0 & 0 \end{pmatrix}$$

Here p_{ij} denotes a positive element. We can see that 4 ↔ 2 ← 1 ← 3 and 4↔ 2 ← 3.

We can see some similarities in Markov chain and CORAS diagram, in sense of that one thing affect the other. In Markov analysis state changes from one incident to another, in other words: Some thing that happens in state1 affects state2 in some way. In CORAS we see how risk, vulnerabilities and unwanted incident have an affect on each other.

We will in this thesis focus on homogeneous and discrete Markov chains as they do not depend on time. There is a possibility that for some large system where non-homogeneous Markov chains could be more appropriate, but we will not focus on that aspect in this thesis. The reason for focusing on homogeneous and discrete is with respect for CORAS diagrams. In CORAS we have been given the states where the transitions will happen, and time is not the most essential aspect of it. Markov chains with use of transitions probability matrix will give us the opportunity to find probability from a threat scenario to an unwanted incident.

[4] A matrix whose entries are non-negative and each of whose rows sum to 1

3. Problem analysis

This chapter will give the main purpose of this thesis, and characterise the problem description. We will go step wise through where Markov chain will be suitable in CORAS diagrams, and try to figure out if that will make CORAS more expressive.

We have seen two other analyse methods, HAZOP and FMECA, and how they work. It is only CORAS which show which vulnerabilities can cause risks using UML, showing the different states, what can occur when something happens. CORAS is also easy to understand for others because it uses pictures and diagrams to describe the scenarios, and how one thing can affect the other.

In CORAS diagrams we analyse the system using four diagrams. This gives us a behaviour perspective on how incidents can happen, who initiates them and what they affect. It is very important to find the right assets in other to make the analysis correct, if not the result of the analysis will be wrong. The asset as described above is something that is valuable for the client, and needs protection. It is the assets that are endangered by the weaknesses that can be subjugated by one or more threats; we call these weaknesses for vulnerabilities. All of these things affect each other, one thing leads to another. If there is risk for something to happen that will have an impact upon the assets, then there are some unwanted incidents that will make this happen. The cause for an unwanted incident is a threat. [7]

Thus risk level is measured by either frequency or probability for the unwanted incident to happen and its consequence. The scale used can be high, medium, low or other scales, and it is called risk value.

The similarity between CORAS and Markov chains tells us that the system will behave differently considering what kind of transaction it gets, like in CORAS, where the behaviour is different from one state to another state. The main aspect here is how to relate Markov chains with CORAS diagrams. The area where this could be done, as we have said earlier, is in step five[5] of CORAS security analysis. Here Markov chains can be introduced to give exact and more elucidate answers. The client will then not have only the CORAS diagram to look at, but a more detailed description of the problem in form of Markov chains.

It is in step five of the CORAS security analysis Markov chains can be altering. We are interested in seeing if it will and if so, how it can give more secure answer when estimating consequences and likelihood values. This is what we will be investigating further in this thesis.

How Markov chains can be introduced in CORAS, is the main question. The minimum we need to know is the likelihood between every state in the threat diagram to try to apply Markov chains. This likelihood is essential for Markov chains. Markov chains can not help us to find the probability for every state in the threat diagram, but it can calculate the total probability from state1 to state2. Will this do CORAS more expressive and the probability estimation more reliable? Today there are probability calculations involved in CORAS which can deliver understandable answers for the probability from state1 to state2. Markov chains must deliver something more then just this, and it possible if we can take full advantage of

[5] Guidelines of how to use CORAS in chapter 2.3

Markov chains in CORAS analysis. The introduction of Markov chains, if possible, will also make CORAS diagram more trustworthy towards other risk analyse methods, since they do not have any calculating methods in range of Markov chains.

Sequence diagrams is a useful way to show how a system works, but such diagrams can also be used to show how CORAS works and how Markov chains will influence the CORAS diagrams. Sequence diagram simply display the lifelines of participating objects as they exchange messages in a single scenario. The life line represents the developing life of the participating object by showing relevant events that are important to the object.

In this example we will only show the threat diagram, and then use sequence diagram to show how it behaves. Then we will introduce Markov chains in it to give an idea of where it is possible to be used.

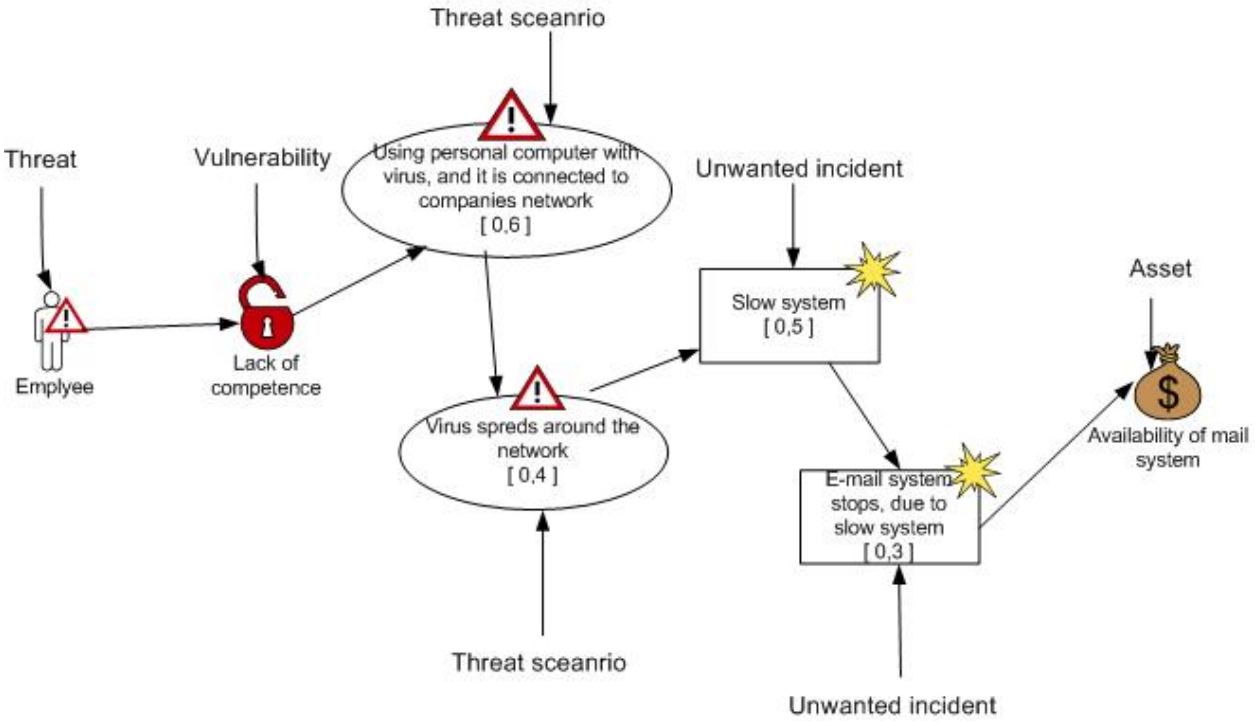


Figure 3-1: Threat diagram

This threat diagram gives an image of a company whose employees do not have efficient competence. The employee uses his/her own personal computer which is infected with virus, and connects the computer to the company network. The virus spreads around the network, which then causes slowness in the system which further makes the e-mail system to stop. All these incidents affect the asset: Availability of mail system. The likelihood given in the brackets of threat scenario and unwanted incident are what the risk analyse team have obtained to. Markov chains do not give the likelihood that is given in those brackets, but it has the possibility to use the likelihood that could be given between the scenarios.

The sequence diagram gives an illustration of how the threat diagram works.

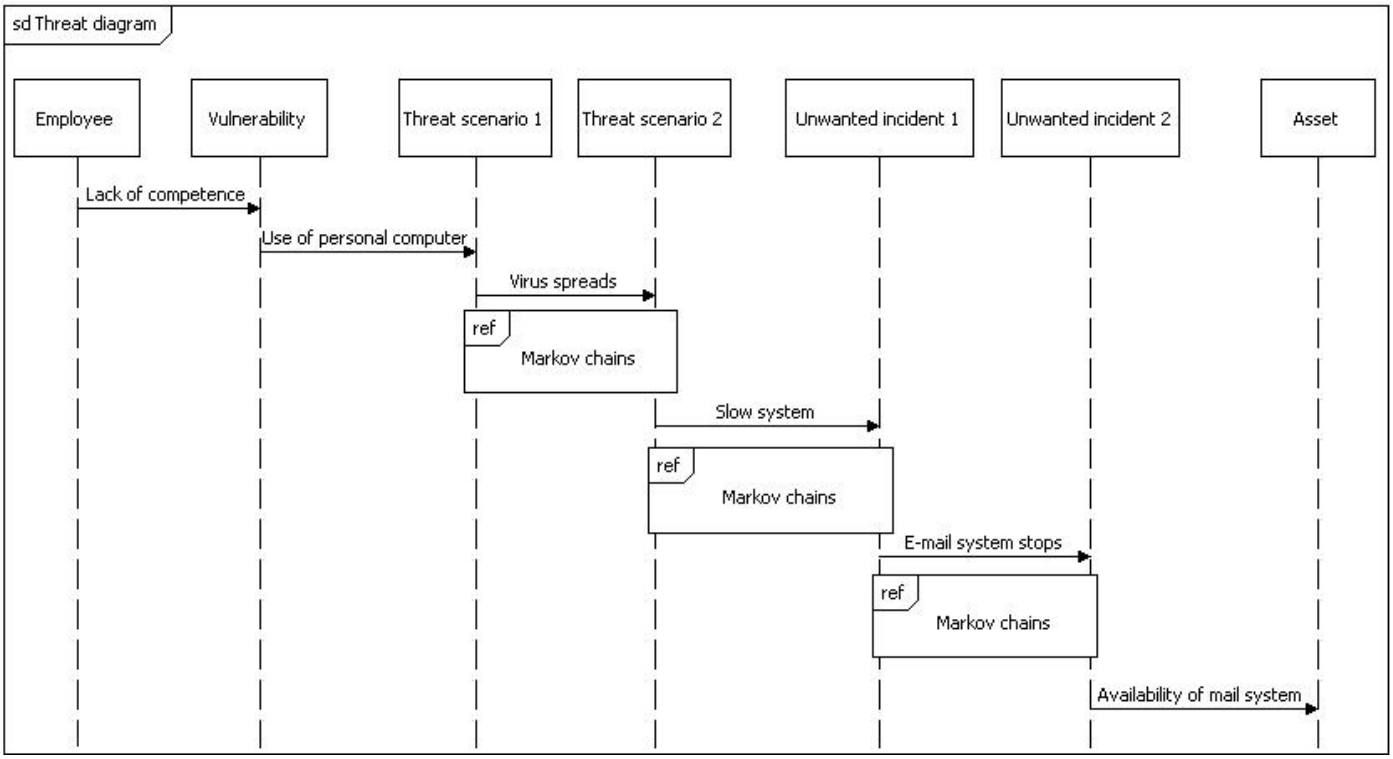


Figure 3-2: Sequence diagram

The square boxes with the name "Markov chains" are where Markov chains can take the given likelihood and use it to calculate the probability

How the system works inside the square boxes is shown below:

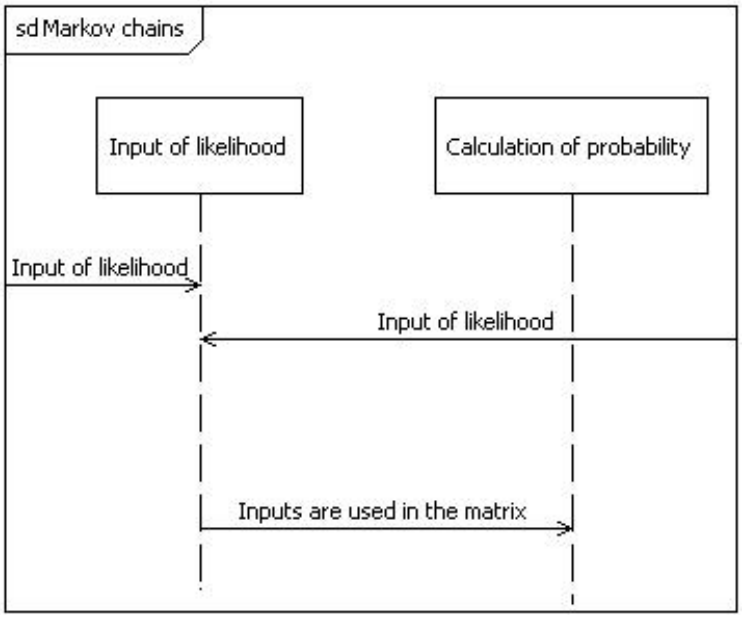


Figure 3-3: Sequence diagram

These diagrams shows us how Markov chains can make CORAS diagram more communicative by its probability calculations, if the full potential of Markov chains are to be used. The original threat diagram has no such advance function of doing so, and this makes CORAS diagrams more reliable with respect to the probability estimation.

The introduction of Markov chains into CORAS diagrams is a new phenomena, and therefore we have no tool-support when introducing it today. We are not going to make a tool that integrates Markov chains to CORAS analysis in this thesis. But to give a picture of how other analyse methods and system have build up a tool, we can se at these links. These tools integrate Markov chains with other methods or systems, and they will give us a picture of in which direction we should be thinking. The tools that are available for using Markov chains are MKV and Relex. MKV is a program for analysing state transition diagrams using numerical integration techniques. For more details MKV visit www.isograph-sowtare.com. Relex uses also state transition diagrams, for more information visit www.relex.com.

The Success criteria for which we will be trying to accomplish during this thesis are:

1. The thesis should compare Markov diagrams to CORAS diagrams with value to:

    - Expressiveness
    - Probability estimation
    - Tool-support
    - Purpose

2. The thesis should deliver a method integrating Markov analysis in risk analysis based on CORAS diagrams:

    - Makes it more reliable
    - Makes it more efficient
    - Gives more benefit on probability estimation

4. Relating CORAS diagrams and Markov chains

In this chapter we will look at CORAS diagrams in all of its phases and explain its purpose. We will also relate Markov chains to CORAS diagrams. We will use the case study to show how CORAS analysis work and try to insert Markov chains to it in chapter 5. This will give a wider depiction of how it affects CORAS diagrams.

The example in chapter 3 is sequence diagram has shown us the main area where Markov chains could be used in threat diagram. With Markov chains we can calculate the probability from a threat to an unwanted incident. This calculation can give more precise and a more trustworthy illustration of the threat that is resting in the system. The probability will be calculated through Markov chains using probability matrix, and any one with math expertise can solve them. The likelihood needed to solve these probability matrixes will be calculated in our simulation tool. This simulation tool is not a part of CORAS-tool, but it will give guidelines of how to operate if we want to relate Markov chains to CORAS. When the analyses team has made the threat diagram, they will have the option to calculate the probability from one state to another. The probability that is plotted in the threat diagram must be accurate. Markov chains can not help to find the probability that is plotted in by the analyse team. Markov chains can only calculate the probability of an occurrence from one state to another, given the likelihood of the paths of both states. The probabilities calculations that Markov chains have the possibility to present are not given by CORAS today. The introduction of Markov chains, if possible, will give many benefits to CORAS in means of being communicative, dependable and giving the firm who wants the risk analysis done by trustworthy calculations. If we can introduce Markov chains here it will have positive affect on other analyse methods to integrate some sort of probability calculating methods.

4.1 Definition of CORAS threat diagram

CORAS diagrams are made step wise and new modelling elements are introduced based on careful consideration.

A threat diagram gives a picture of what can have consequences on the assets, and to give this illustration it uses elements as threats, vulnerabilities, threat scenarios, unwanted incidents, and assets. These building blocks are illustrated in figure 2. Threat diagrams have three different relations: *initiate*, *leads to* and *impact*. The *vertices* of the threat diagram are threats, threat scenarios, unwanted incidents, and assets. It is also possible to allocate likelihood to threat scenarios and unwanted incidents likelihood. [15]

The semantics of how a threat diagram can behave in different ways can be explained in this textual syntax [15]:

diagram = *({vertex}⁻, {relation});*
vertex = *threat | threat scenario | unwanted incident | asset;*
relation = *initiate | leads to | impact;*
initiate = *threat* $\xrightarrow{\text{[vulnerability set] [likelihood]}}$ *threat scenario |*
threat $\xrightarrow{\text{[vulnerability set] [likelihood]}}$ *unwanted incident;*
leads to = *threat scenario* $\xrightarrow{\text{[vulnerability set] [likelihood]}}$ *threat scenario |*
threat scenario $\xrightarrow{\text{[vulnerability set] [likelihood]}}$ *unwanted incident |*
unwanted incident $\xrightarrow{\text{[vulnerability set] [likelihood]}}$ *threat scenario |*
unwanted incident $\xrightarrow{\text{[vulnerability set] [likelihood]}}$ *unwanted incident;*
impact = *unwanted incident* $\xrightarrow{\text{consequences}}$ *asset | threat scenario → asset;*

threat = *deliberate threat | accidental threat | non-human threat;*
deliberate threat = *identifier;*
accidental threat = *identifier;*
non-human threat = *identifier;*
vulnerability set = *{vulnerability}⁻;*
vulnerability = *identifier;*
threat scenario = *identifier [(likelihood)];*
unwanted scenario = *identifier [(likelihood)];*
asset = *identifier;*
likelihood = *linguistic term | numerical value;*
consequence = *linguistic term | numerical value;*

The likelihood given in every element of the threat diagram is decided by brainstorming or based on historical data. In unwanted incidents the likelihood is either determined through brainstorming or by calculating the given likelihood in the other elements.

4.2 Definition of Markov chains

Markov chains are used to calculate the probability of sequences, from one state another. When using Markov chains it is significant that we know the likelihood for each state to occur, without this we can not calculate the probability for the given sequence.

There are three different paths, which are suitable for CORAS threat diagrams, from where the likelihood can be used in Markov chains, that is from one state to another, the likelihood for staying at the same state (absorbing), and the last is the way back to a previous state (reverse). The figure below gives an illustration of this:

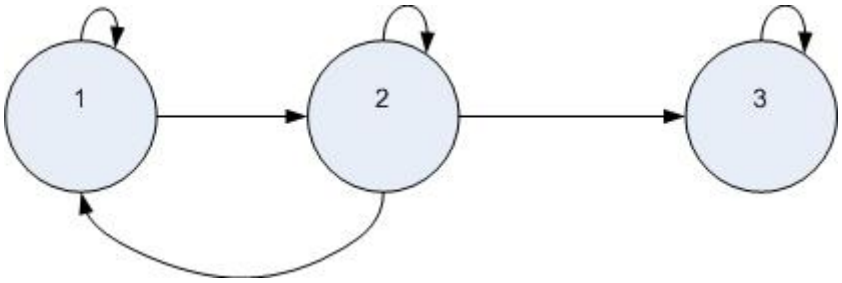


Figure 4-1: The possible cases of CORAS threat diagrams

The likelihood from a state to another must be given in the path, in other words: likelihood will tell us that this is the possibility from this state to another.

When calculating the probability it is essential to know the amount of steps used from one state to another. This amount of steps decides how the matrix will look like, and what calculation result will be.

4.3 Does CORAS threat diagram and Markov chains have common characteristics, and is there any limitations?

We can think of Markov chains as a set of states, like a state diagram. We make this state diagram with respect for threat diagram as we convert the threat diagram to a state diagram. This state diagram gives a broader image of the means of Markov chains, and how they work. The state diagram shows step wise in which order the different scenarios can occur, and what the given likelihood for that is. Then Markov chains can use the given likelihood to calculate the probability for the whole sequence, or the probability from one state to another.

As we have mentioned earlier the likelihood given in a threat diagram or the path from one element to another is determined through brainstorming or historical data, and even if the semantics above shows that it is possible to have likelihood at a given element it is not for sure that likelihood will be given in the threat diagram. We can not always get all the likelihoods that are possible to accumulate in a threat diagram through brainstorming. The likelihood given in a threat scenario or an unwanted incident does not always have to be in numerical expression, it is possible that the likelihood is given in a linguistic value like [low], [medium], or [high], it is also possible that it is given as [1 per year] or [1 per 6 months]. We know that it is possible to make a numerical scale in the CORAS tool for the linguistic values; this scale varies from system to system. This scale can not be used in Markov chains because there we need precise likelihood.

We will further in this chapter see if it is possible to start applying Markov chains to a threat diagram, and what can be done when the likelihood between the elements is not given. How to apply Markov chains when the given likelihood is only in the threat scenarios and unwanted incidents. Are there any common characteristics and are there any limitations? And are these likelihoods enough to take full advantage of Markov chains?

The idealistic threat diagram is where the likelihood between all of the elements is given through brainstorming, and it is in numerical value. When this likelihood is given, there is no problem if the likelihood in the threat scenario and unwanted incidents are of linguistic value or numerical expression. We will see if Markov chains can only depend on those likelihoods given between the elements.

We will transform the threat diagram from figure 3 into state diagram:

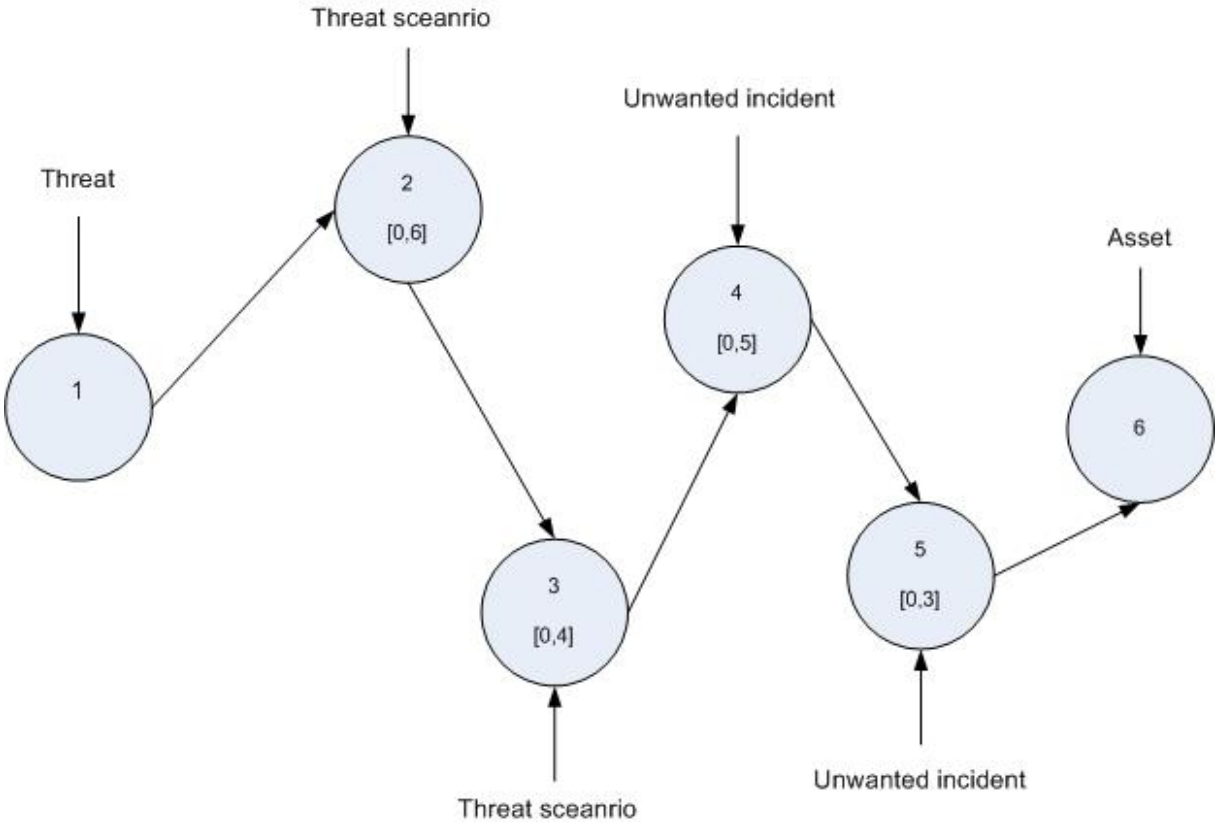


Figure 4-2: State diagram illustrating the transformation of the threat diagram in figure 3

We see that state1 represents the threat, state2 is the threat scenario etc. The state diagram moves step wise further for each step. This state diagram is a direct transformation from the threat diagram; we can see the likelihood given in those brackets inside the states.

The vulnerability elements from the threat diagram are not included in this state diagram. To understand this we have to look at the semantic of CORAS threat diagram. The semantics tells us that the vulnerabilities is not a "state" for it self, but indicates that the threat exploits the vulnerability to initiate a threat scenario or an unwanted incident. We interpret from the textual explanation that: *threat* $\xrightarrow{\text{[vulnerability set] [likelihood]}}$ *threat scenario*. The graphical explanation is:

Figure 4-3: Initial relation

This is the reason that the vulnerabilities are not to be looked as a state when transforming a threat diagram to a state diagram.

Some will maybe suggest that the next step will be to transform this stated diagram into this:

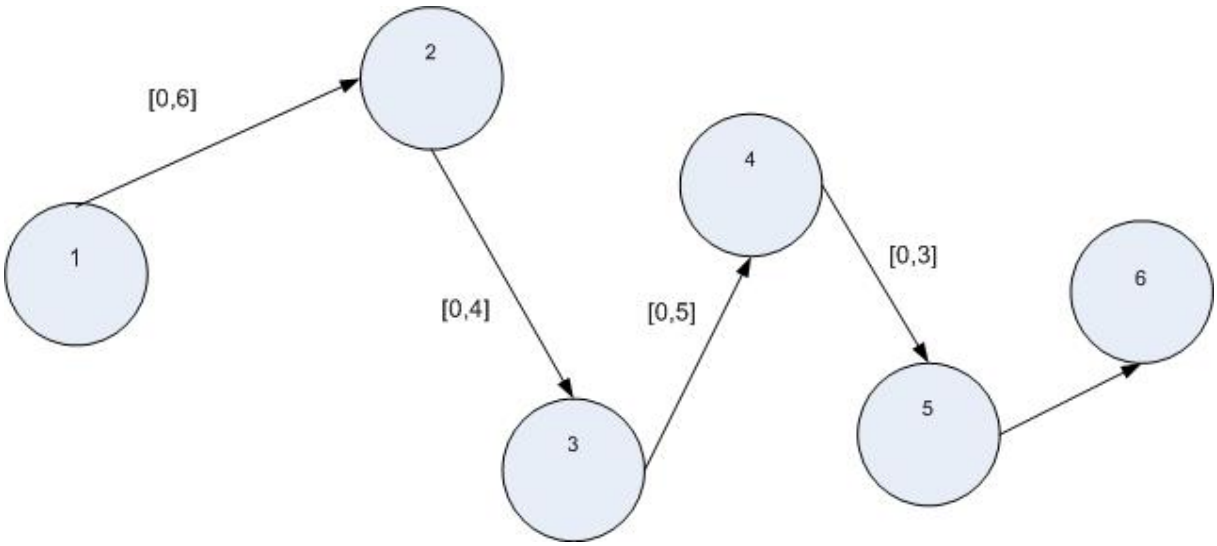


Figure 4-4: State diagram illustrating the threat scenarios

How can we take the likelihood given in the state, and put it outside to show that this is the likelihood for this state to occur?

The semantic of CORAS threat diagram gives us these *vertices*: [15]

*[[dt]] := **dt** is a deliberate threat*
*[[at]] := **at** is an accidental threat*
*[[nht]] := **nht** is a non-human threat*
*[[a]] := **a** is a asset*
*[[ts]] := Threat scenario **ts** occur with undefined likelihood*
*[[ts(l)]] := Threat scenario **ts** occur with [[l]]*
*[[ui]] := Unwanted incident **ui** occur with undefined likelihood*
*[[ui(l)]] := Unwanted incident **ui** occur with [[l]]*

A threat scenario can either occur with an undefined likelihood or with a known likelihood.

The semantic shows also the possible initiate relations, which are: [15]

1. *[[t ⟶ ts]] := **t** initiates **ts** with undefined likelihood*
2. *[[t $\xrightarrow{l}$ ts]] := **t** initiates **ts** with [[l]]*
3. *[[t $\xrightarrow{V_n}$ ts]] := **t** exploits [[V_n]] to initiates **ts** with undefined likelihood*
4. *[[t $\xrightarrow{V_n\ l}$ ts]] := **t** exploits [[V_n]] to initiates **ts** with [[l]]*

We can see that in our example we are at initiate relation 3, our threat exploits a vulnerability to initiate a threat scenario and the likelihood is undefined. We can from the vertices and our threat diagram see that we have a threat scenario which occurs with a likelihood of 0,6. The likelihood l_1 as shown in the graphical explanation is unknown.

We have to realize that the CORAS semantics does not allow taking out the likelihood given in an element and show it as the paths likelihood. If this is the case we can not use Markov chains to calculate the probability. We have to go back to the brainstorming phase and find these likelihoods, and if this is not possible then we have no possibility to apply Markov chains directly to a threat diagram.

When applying Markov chains to calculate the probability from one state to another, or for a whole sequence, Markov chains tries every possible way from the first state to the last one. Markov chain allows a state to remain in its given state (absorbing condition), and the likelihood given here can be used to calculate the probability. This is counted as one step in the calculation. Markov chains gives also a state possibility to have a relation back to previous state (reverse condition), the given likelihood for this is also used when calculating the probability.

These two aspects are shown in the two state diagrams below:

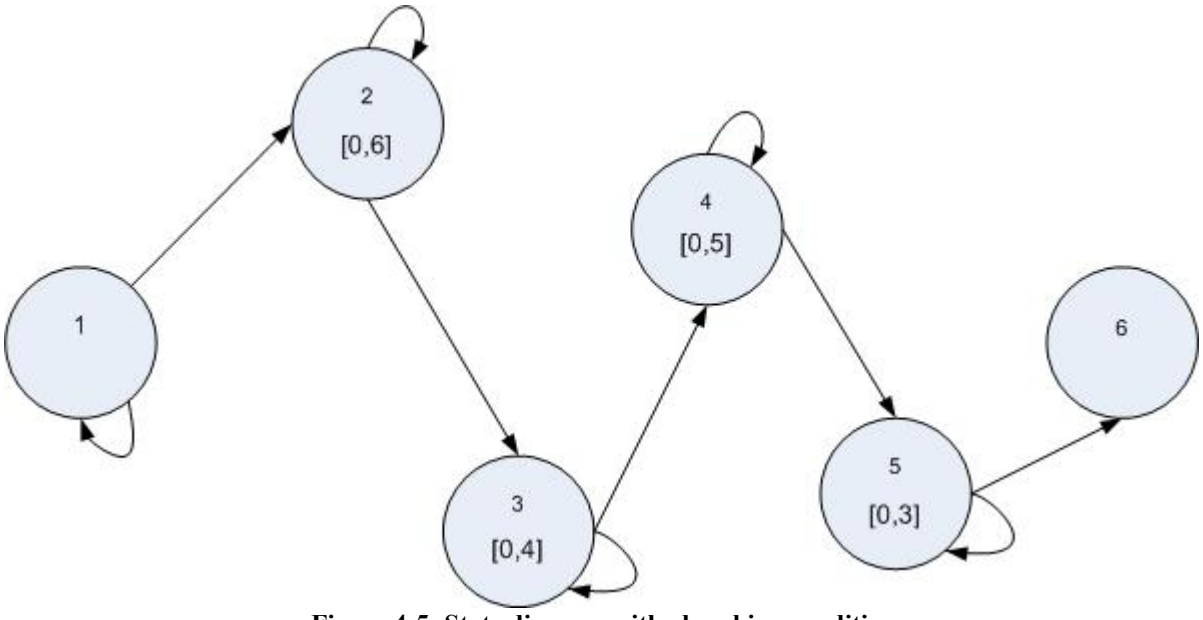


Figure 4-5: State diagram with absorbing conditions

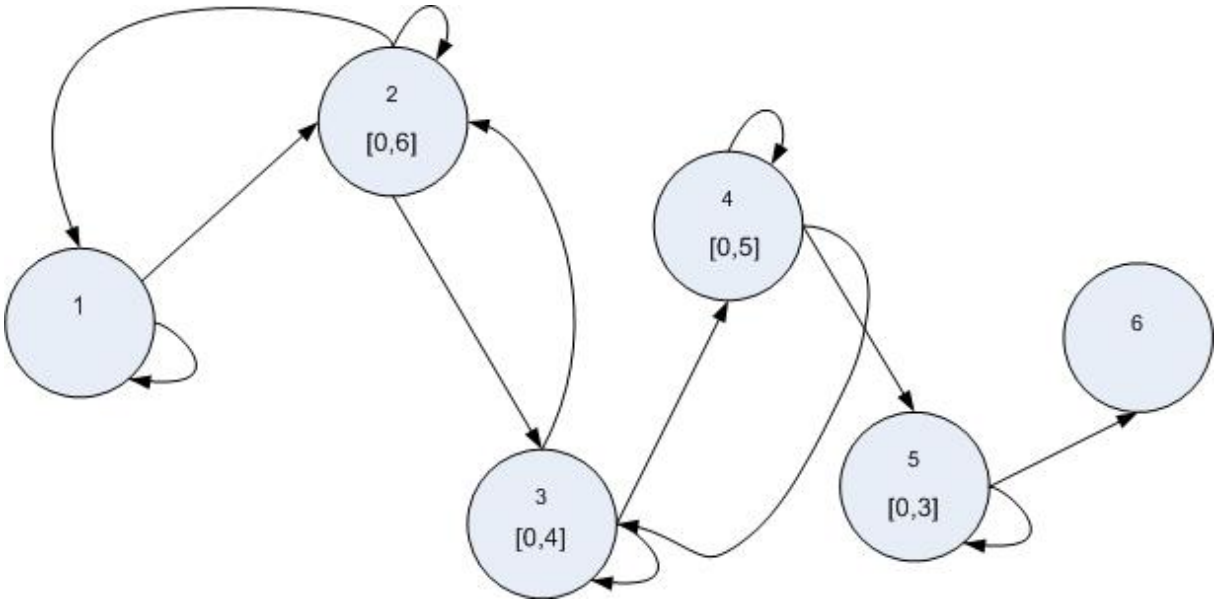


Figure 4-6: State diagram with both absorbing and reverse conditions

The likelihood that Markov chains uses from these two aspects are also not to be found in CORAS threat diagrams, because these aspects are not a part of the threat diagram. There are two questions here:

- Does the CORAS threat diagram semantics allow us to introduce these two features in the threat diagram?
- If these features are not included, will it affect the calculation in Markov chains?

We know from the CORAS threat diagram semantic that every element points in one direction.

initiate = *threat* —[vulnerability set] [likelihood]→ *threat scenario* |
threat —[vulnerability set] [likelihood]→ *unwanted incident;*

We go from one state to another; none of the states preserves them in the same state or goes back to the previous. So this is not possible to insert in the CORAS threat diagrams.

If we suppose that we can introduce a state that can preserve in the same state in threat diagram, how would this affect the likelihood that is given in the state from before? And how can we calculate the likelihood for the state to stay in the same state.

We know that it is a possibility that there are some other unknown paths which is not a part of our threat diagram. So the threat diagram we present has likelihood for not including some aspects that could occur. This gives those unidentified elements for occurring likelihood that is unknown for us.

Example 4-1:

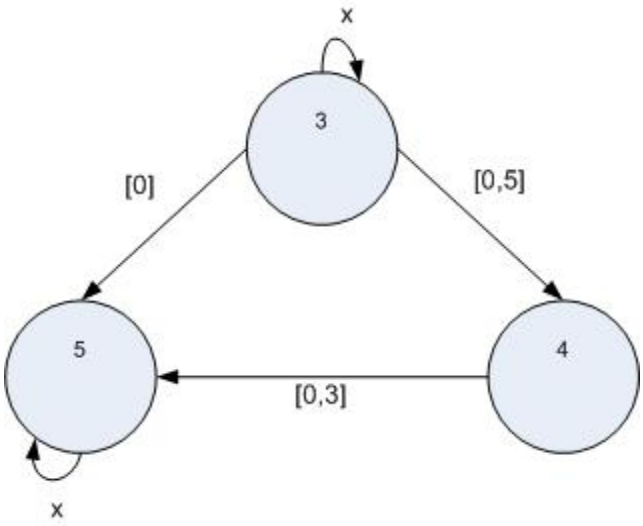


Figure 4-7: A part of a state diagram (figure 4-2)

This is a part of the above state diagram, figure 8, with some modifications which will be explained. We choose the transaction from state 3 → state 5, and try to apply Markov chains to find out the probability for the transaction going from state 3 to state 5. The **x** here represents the unknown likelihood for a state to stay at its state. We have hypothetical given the likelihood between these three states.

As we can se from this figure we have different paths from state 3 to state 5 and there is a reason for this. We mentioned earlier that CORAS diagrams only goes in one direction, and the reverse transaction is not possible. Another thing that is important is that the relation between the different states is static. When reading the threat diagram we must follow the path that is written in it, that we for example go from a vulnerability to a threat scenario and that goes further to an unwanted incident. It is not possible to assume that we can skip one step and go directly from a vulnerability to an unwanted incident. Neither can we assume that if there are two threat scenarios linked to each other and the last threat scenario is linked to an unwanted incident, and we can skip one threat scenario to get to the unwanted incident. These rules are static in CORAS diagrams, we must follow the given steps from one state to the next. Then why do we have a relation between state 3 and state 5? This is to give a clear picture how Markov chain calculates probability. Markov chain must try every option possible to go from state 3 to state 5, even if it is not given in the threat diagram. Since this relation is not a part of the threat diagram, we have given it likelihood 0.

Another thing we comprehend from threat diagram that must be used when calculating the probability is the amount of steps taken from state 3 to reach state 5. We look at the threat diagram, figure 5, where it is used 2 steps from the given threat scenario to an unwanted incident. We use the same amount of steps when calculating the probability using Markov chains in this example. When we analyse figure 13 we must use the same amount of steps to

get from state 3 to state 5, as it is made in the threat diagram. There are three different paths from state 3 to state 5 are:

- State 3 remains at state 3 and then goes to state 5
- State 3 goes to state 4 and then to state 5
- State 3 goes to state 5 and state remains there

We construct a probability matrix by looking at the possible paths in the whole scenario. Since we do not know the likelihood for the different combination Markov chain has the possibility to evaluate, we have inserted the value 0 there. The probability matrix does not include state 6. The reason for this is that state 6 is the asset we are trying to protect, and that is the reason it is a part of the matrix.

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} x & 0 & 0 & 0 & 0 \\ 0{,}6 & x & 0 & 0 & 0 \\ 0 & 0{,}4 & x & 0 & 0 \\ 0 & 0 & 0{,}5 & x & 0 \\ 0 & 0 & 0 & 0{,}3 & x \end{pmatrix} \end{array}$$

To calculate the probability for state 3 → state 5 we will be only using the matrix from 3 to five. The equation that is used in Markov chains is:

$$P_{ij} = \sum_{k=1}^{r} P_{ik} P_{kj}$$

The result is:

$P_{33}P_{35} + P_{34}P_{45} + P_{35}P_{55} \Rightarrow x*0 + 0{,}5*0{,}3 + 0*x = \underline{0{,}15} \Rightarrow \underline{\textit{15\%}}$

There is 15% probability that the scenario will go from state 3 to state 5. This calculation helps to give a more reliable answer of how big threat this is. But have we used the full potential of Markov chains? What we have done here is just multiplied the different likelihoods with each other. This could be done using any normal probability calculating method, by assuming that we actually do not need to suppose the absorbing likelihood. We see that all other paths then those given in the threat diagram multiply with 0. We have to make some changes if we want to use Markov chains on CORAS threat diagrams. If those features that are given in Markov chains are not included, then the calculation above is the result. We can see that we do not need Markov chains to calculate this straight forward multiplication.

Before we see what is possible to do if Markov chains is to applied to CORAS, we will look at some other aspects in threat diagram where some interpretation has to be done before Markov chains can be applied.

CORAS threat diagram semantic confirms that that one or more threats or threat scenarios can be pointing on the same threat scenario or an unwanted incident. Let us look at this threat diagram:

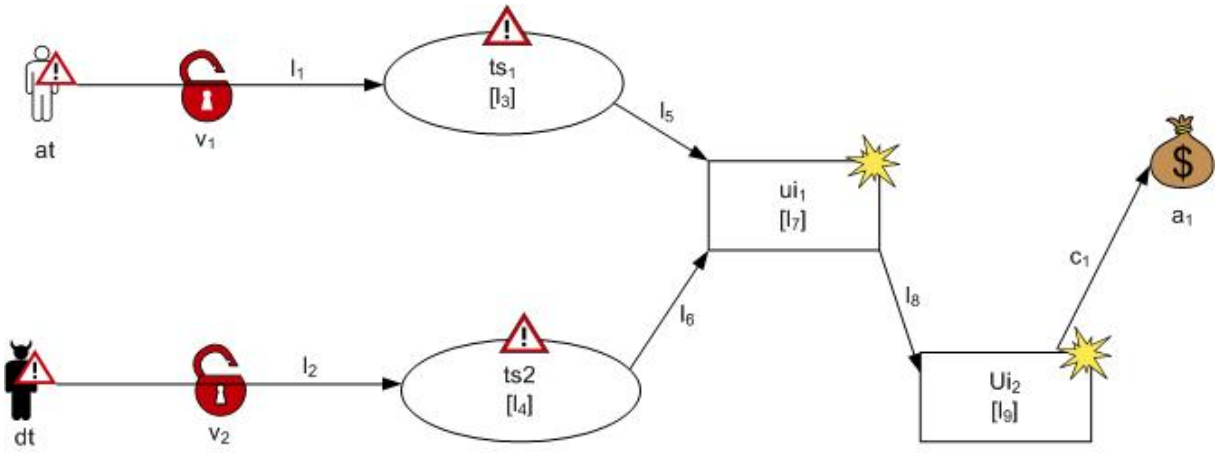


Figure 4-8: Threat diagram

We see that both ts_1 and ts_2 points to ui_1. We se that the likelihood from *at* → *ts1* is l_1 and that l_2 is the likelihood from *dt* → *ts2*. The likelihood from ts_1 → ui_1, called l_5, and the likelihood from ts_2 → ui_1, called l_6, is unknown. We know from earlier studies in CORAS that if we want to calculate the likelihood given in ui_1, l_3, we can estimate it by using this equation [18]:

$$l = (l_1 \times p_1) + \ldots + (l_m \times p_m)$$

The **p** given in this equation is the likelihood in the paths.

This equation is used when brainstorming or historical data determine the p_1 and p_2 but does not conclude what l is going to be.

There are three scenarios that may come up:

1. We have the "ideal" threat diagram for Markov chains, where all likelihoods that are possible are given.

2. Either l_5 or l_6 is given in the threat diagram during the brainstorming process. We assume here that all possible likelihood is given, expect for l_6. Then we can use this equation to find the unknown l_6. If we suppose that

   $l_3 = 0,6$, $l_4 = 0,3$, $l_7 = 0,5$, $l_5 = 0,4$ and l_6 = **unknown**

   Then we can calculate l_6 by transforming this equation to:

$$l_6 = (l_7 / l_3) - ((l_4 \times l_5)/ l_3)$$
$$l_6 = (0,5/0,6) - ((0,3x0,4)/0,6)$$
$$\underline{l_6 = 0,63}$$

   We see that we find l_6 by solving this equation.

3. If we assume that l_5 and l_6 are unknown in this equation then we can not solve it. It is not possible to approach this problem with a mathematical equation since neither l_5 nor l_6 in known.

There are two options that could be done if scenario three is to occur:

- We have to go back to the brainstorming and historical data part of the analysis and try to find out one of the unknown likelihoods, if it is possible by looking at all of the facts. Then we can use the equation given above to find the other unknown likelihood.

- If the above option does not give any results, we can not calculate the probability. We must do some adjustments if we want to calculate the probability. What these adjustments are we will be looking on further in this thesis.

If we have an ideal threat diagram, a threat diagram where all of it its likelihoods are given, then how can Markov chains be applied? We know now that we have to introduce the absorbing condition and the reverse condition if we want to have full benefit of Markov chains. Threat diagrams are just a part of the whole system; it does not give the real picture of how a system is built up and how it works. We have to collect more data from the brainstorming face and from the historical data to use Markov chains on CORAS threat diagrams. We need some experimental data, like measurements, and simulate the system to get the values. This experimental data and simulation will differ from system to system, so we need to know which data is needed for Markov chains. To this we have build state diagrams, and if the historical data and brainstorming process does not produce the needed data, we have to execute some simulations.

The state diagrams used in this process resemble those used when transforming a threat diagram to a state diagram, but we have to make some changes. We show all those paths that are needed for Markov chains with respect for the CORAS threat diagram, and make state diagram for each sequence. The relation between a threat and a threat scenario must be looked as one sequence and further on. This has to be done because we have to simulate each given element and its relation to the next element, because this is the best way to collect the values we need. This simulation will also test the likelihood given in the paths in the threat diagrams. Does the given likelihood from historical data or brainstorming resemble with the likelihood calculated from the simulation? This will show good our simulation is. This simulation helps with another thing, if the paths likelihood is not given in the threat diagram, then we can simulate us to find it.

Let us look at this threat diagram:

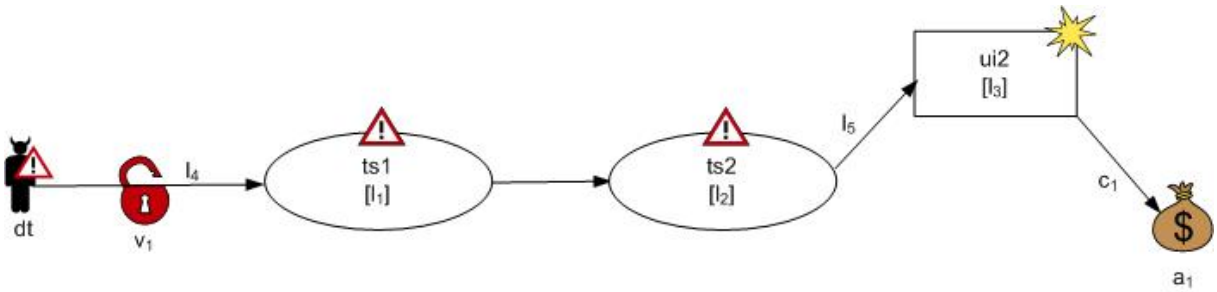


Figure 4-9: Threat diagram

We first transform this into a state diagram with respect to Markov chains.

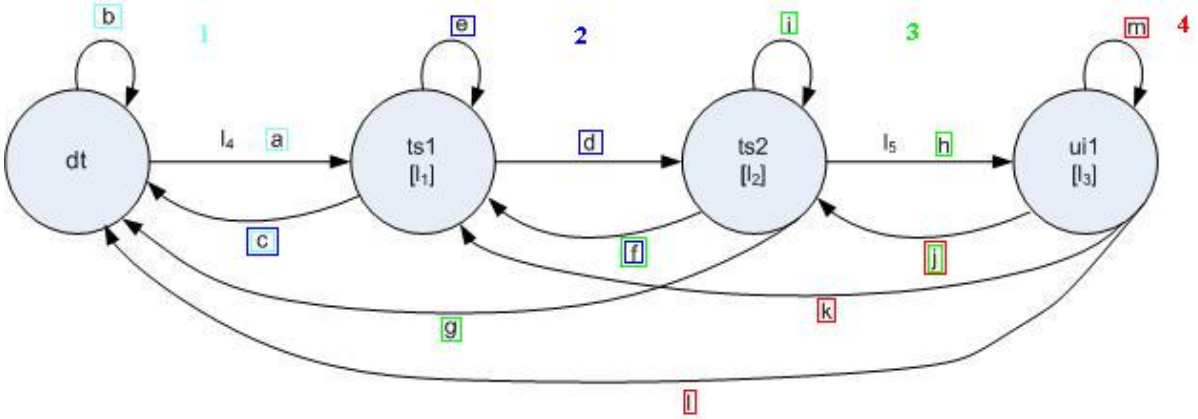


Figure 4-10: State diagram transformed from the threat diagram (figure 4-9). Illustrating the different needed likelihoods and how the diagram should be divided

This state diagram is not just a direct transformation of the threat diagram, but it includes the likelihoods that are needed for Markov chains. Since this is the diagram we will be testing our system on, we have not taken the asset in it. The asset is not part of the system, it is what we are trying to protect. The deliberate threat is also not part of the system, but it has to be included because this is the source for this sequence to happen. We see that we have pointed out how we have to simulate every section number wise. This is because we have to calculate the likelihood for each section to get the correct answer.

We start by looking at the first section:

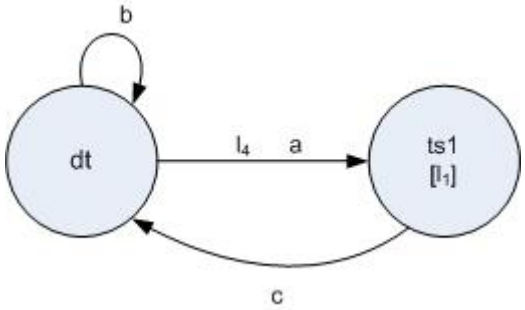


Figure 4-11: First section of the state diagram

We have called the absorbing likelihood back to *dt* for **b**. The likelihood from *dt* → *ts1* for **a**, and the reverse likelihood from *ts1* → *dt* for **c**. We know that $a = l_4$, but we will test this also out when simulating this scenario. Then we can se well the simulation went. Before we start to simulate it is essential to look at the historical data and brainstorming phase. We have to make some assumptions before we can start simulating; these assumptions are made by going back and looking at the historical data and brainstorming phase. The assumption sets a maximum limit for an incident to happen, and we use this maximum limit when we simulate. The maximum limit tells us that a given incident can not occur more then the maximum limit, it can occur less then the given limit. We make new assumptions for each time it is needed. The assumption are essential in this simulation, with out it we can not simulate to find the likelihood. These likelihoods are needed if we want to apply Markov chains to CORAS threat diagram.

For instance if we think that our **dt** resembles *a hacker*, $\mathbf{v_1}$ is *poor firewall* and **ts1** resembles *hacks into companies network*. Then we have to look at the historical data and through brainstorming determine for example *how many hacks is possible per day*. This amount will then be the limit set by the analyse team for the simulation for this case. These assumptions must be made for each new sequence we are simulating.

The rules for how to calculate the likelihood after simulating:

a: is the likelihood for *state dt* → *state ts1*. We must use the assumption made that sets a limit for an incident to occur. We then simulate this for a period of time, which is also determined throw brainstorming and historical data. We then add the answer we get from each attempt and divide it on the time period. This gives us the average, which is the likelihood for this event to occur. We can write this calculation as:

$$m = (f_1 + f_2 + \ldots + f_n) / N_f$$

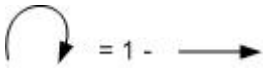
m is the average

f_n is the result of each attempt

N_f is the time period determined by the analyse team

b: is the absorbing likelihood for number of times it was not possible to go from *state dt* → *state ts1*. We calculate **b** by subtracting 1 from **a**:

$$b = 1-a$$



c: is the reverse likelihood for number of times we got from *state dt* → *state ts1*, but was "thrown out" from *state ts1* → *state dt*. The reason can be a security mechanism there that does this. We have to make a new assumption to calculate this likelihood. This assumption can not be greater then the assumption made for calculating **a**. It can be less or equal to the assumption for calculating **a**. We follow the same procedure as in **a**, and in the end find average by using the calculation method mentioned above.

Section 2:

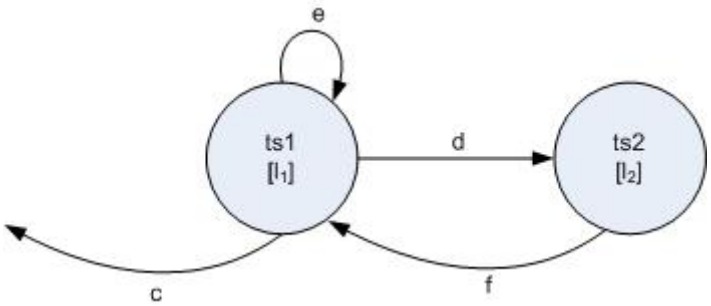


Figure 4-12: Second section of the state diagram

d: is calculated as shown in **a**, but with a new assumption.

e: this calculation is different from the one shown in section 1. Here we have to look at the previous likelihood, c in this case, also. In section 1 there was no reverse likelihood like here, so it was straight forward to calculate it. What **e** resembles here is; the absorbing likelihood for not going back to previous state and not going forward to the next state. We calculate this by subtracting 1 from the likelihood given in **d** and reverse likelihood, **c**, from section 1.

$$e = 1 - d - c$$

We have to remember that if this calculation gives an answer which is in negative format, then something is wrong. Maybe our simulation did not go as planned, or our conjecture was not correctly determined. We then have to go back and see where the fault lays, and try to simulate again.

f: is calculated in the same manner as **c** in section 1. The assumption here can be equal or less then the assumption made for calculating **d**.

Section 3:

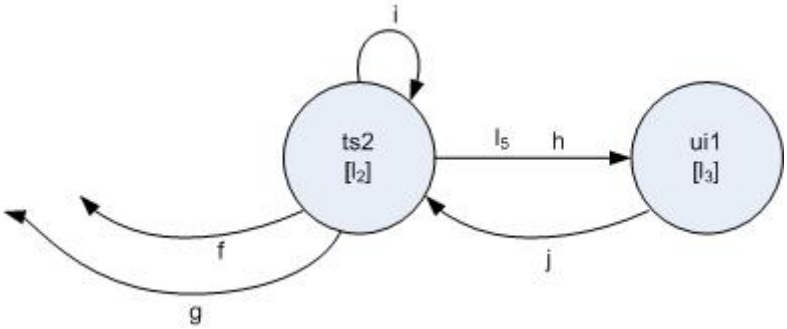


Figure 4-13: Third section of the state diagram

h: is calculated as shown in **a**, but with a new assumption.

i: this calculation is different from the one shown in section 2. Here we have to look at all of the reverse likelihoods, f and g in this case. What **i** resembles here is; the absorbing likelihood for not going back to any of the previous states and not going forward to the next state. We calculate this by subtracting 1 from the likelihood given in **h** and reverse likelihood from section 2 and likelihood for **g**.

$$e = 1 - h - f - g$$

If this calculation also gives an answer which is in negative format, then something is wrong. Then we have to go back and se where the fault lays, and try to simulate again.

j: is calculated in the same manner as **c** in section 1, but with a new assumption.

g: is calculated by using an assumption, as done in **j**, but with a new assumption. It is important here to remember that **g** is the likelihood for the system going back to the first state. This depends on the system, how well it is built and protected. It possible this transaction is not possible in some systems, and then the likelihood is 0.

Section 4:

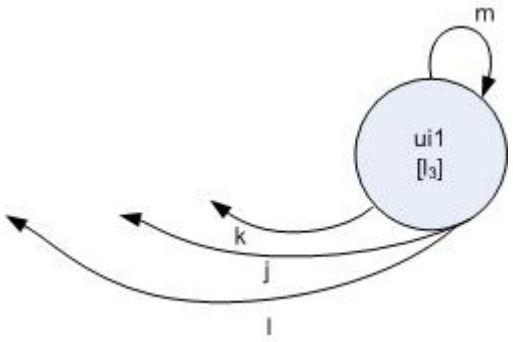


Figure 4-14: Fourth section of the state diagram

The last element requires some other interpretations. We know that this is the last element and it can not point further on to other elements. To find the likelihood of **m** we have to use all of the reverse likelihoods pointing from this state, **i**, **j**, **k,** in this case, and subtract it from 1. The answer of this tells us that this is the likelihood for staying at this given state, and we subtract away that part of the likelihood when it is sent back to the previous state.

$$m = 1 - i - j - k$$

These calculations from the simulation give us the opportunity to take advantage of Markov chains ability.

We have earlier mentioned that when applying Markov chains to CORAS threat diagrams the amount of steps is a major factor. The amount of steps that is to be counted from one state to another state lays the foundation for the calculation of the probability. We have earlier seen that when we take same amount of steps that are used in the threat diagram, and when we apply Markov chains to it, we only multiply the given likelihoods. To get the full benefit of Markov chains we must try more then those steps given in the threat diagram. We know that when we simulate we produce more paths then given in the threat diagram. We have to use them to get the full benefit of Markov chains. We let the given amount of steps in the threat diagram be the minimum amount of steps, and the maximum is unlimited. We do have to remember that we have to follow the semantics of CORAS threat diagram when going from threat to an unwanted incident. Even if it is Markov chains have the possibility to drop some elements and point directly to the last element, we can not do that. This is because we presume that that the given scenario and its path from the first element to the last is fixed. It maybe possible for state in our simulation to point back to a previous state or to stay where it is, but it is not possible to skip one state and point to the next when. This is to preserve the idea of CORAS threat diagrams, that this is the way some thing can occur that can affect our assets. Because we simulate the system with CORAS threat diagrams as our basis.

We will through the case study and risk analysis done on it with CORAS diagrams show how CORAS works. In the next chapter we will use this case study to develop a method for applying Markov chains to CORAS diagrams using the simulation tool we have developed.

4.4 Case study

The reason behind this case study is to give an example of how a risk analysis can be done by using CORAS diagrams. What kind of information one should look for and how to use it in the diagrams. The way of presenting the diagrams and in which way they should follow is described earlier in the paper. We start with collecting the vital information for the analysis, and build the rest of the analysis on this information. The next stop is to create the diagrams; First we create an asset diagram which gives a picture of what is of value that needs protection. The second diagram is a threat diagram, this analysis what or whom can cause harm to the assets. Risk diagram is the second last diagram, here we estimate the consequences. The last diagram is a treatment diagram; here we give a solution for the risks.

Post-Service is a mail-order company that does not have storage of itself. It is therefore dependant on continuous communication with their suppliers, each time they receive a new order from a customer. Customer files and records are considered critical information, which should naturally be kept out of sight of various competitors.

The company requires confidentiality in order to maintain its marketing plans out of reach and as mentioned earlier, out of sight of potential intruders/ competitors. The last mentioned, to prevent outsiders from copying ideas etc. The input of harmful data, false orders or press releases that can lead to negative sanctions for the company is served and protected by the integrity. One of the other essential factors for the company is the availability for their e-mailing system which is integrated with the customer service and CRM-platform.

Firewall is also applied as a Front-end server for e-mailing, as well as it scans incoming and outgoing e-mails for viruses. The customer system is integrated into a CRM system with e-mail connection. When an order is received by e-mail, the e-mailing system automatically alerts the CRM system. At the same time the order is manually registered in the CRM system. Initially a booking of the desirable merchandise is been made to the supplier with the help of e-mailing services, is followed by an invoice to the customers. When the goods are received, they are first repacked and then sent to the customer.

The analysis shows that the there are some elements is the company which can harm the assets, and what kind of risk they can develop. IT-infrastructure, mainly the hardware, is where threats can cause vulnerabilities to the system. There are also external threats, as for instants hackers. The main vulnerabilities which can create a threat scenario and further on develop more vulnerability which can cause unwanted incidents are: hardware error, TCP\IP stack and weakness in the OS.

Hardware error can stop the firewall from functioning, which will distress the mail-server. The orders will not get through and everything integrated with it will be affected.

TCP\IP stack vulnerability has a dynamo affect, it creates another sort of vulnerability in the system. It can overload the firewall and create denial-of-service attack.

Weakness in OS can be exploited by a hacker. He or she can make use of the weakness in OS to take control over the firewall. This can develop into two new vulnerabilities, *shutting down the firewall* which will affect the mail-server, or *using the firewall to further intrusion*. The hacker can then monitor the traffic from inside and get inside information. This will give the

hacker opportunity to reveal the plans the firm has, or the opponents can take advantage of the situation because this is bad publicity for the firm.

The treatment for these vulnerabilities are updating hardware, improve the capacity of the firewall and start using firewall software that controls sessions and using strong passwords and installing an IDS system.

CORAS Risk analysis:

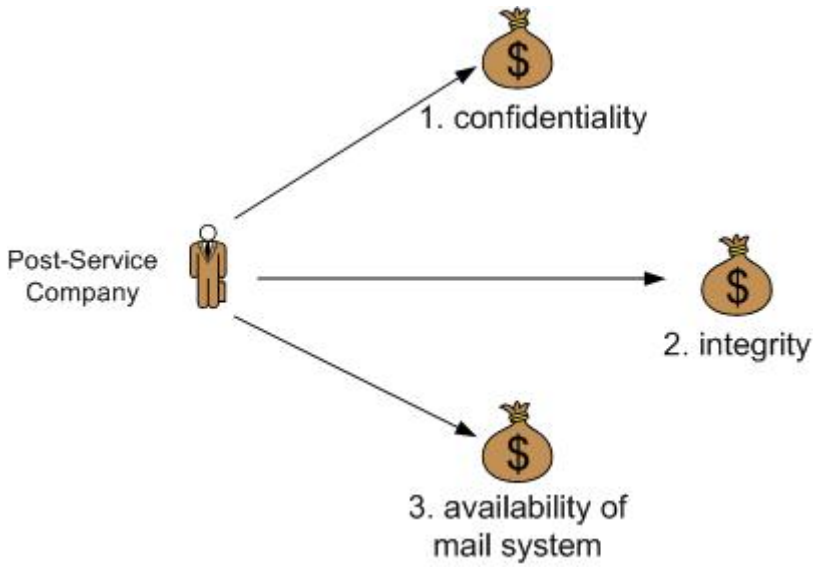


Figure 4-15: Asset diagram (case study)

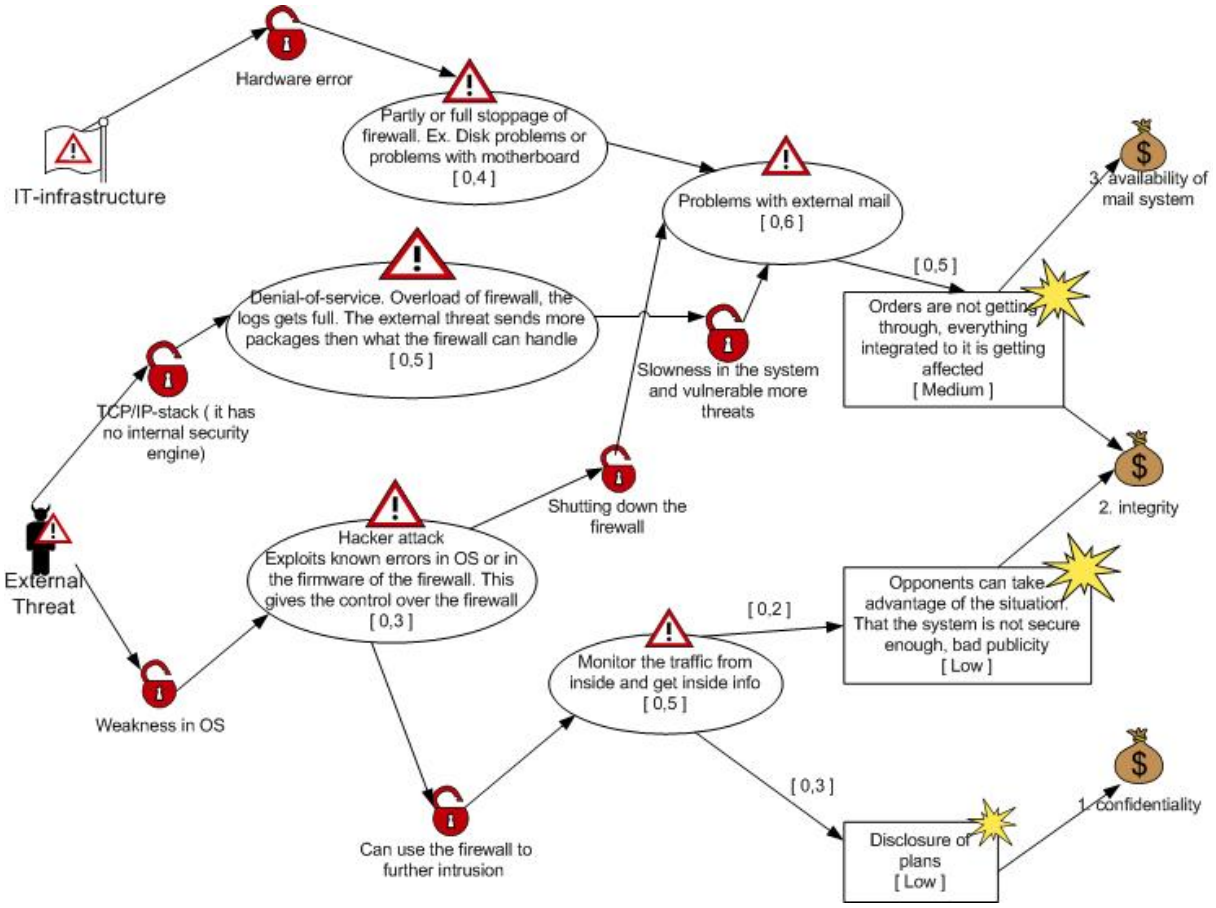


Figure 4-16: Threat diagram (case study)

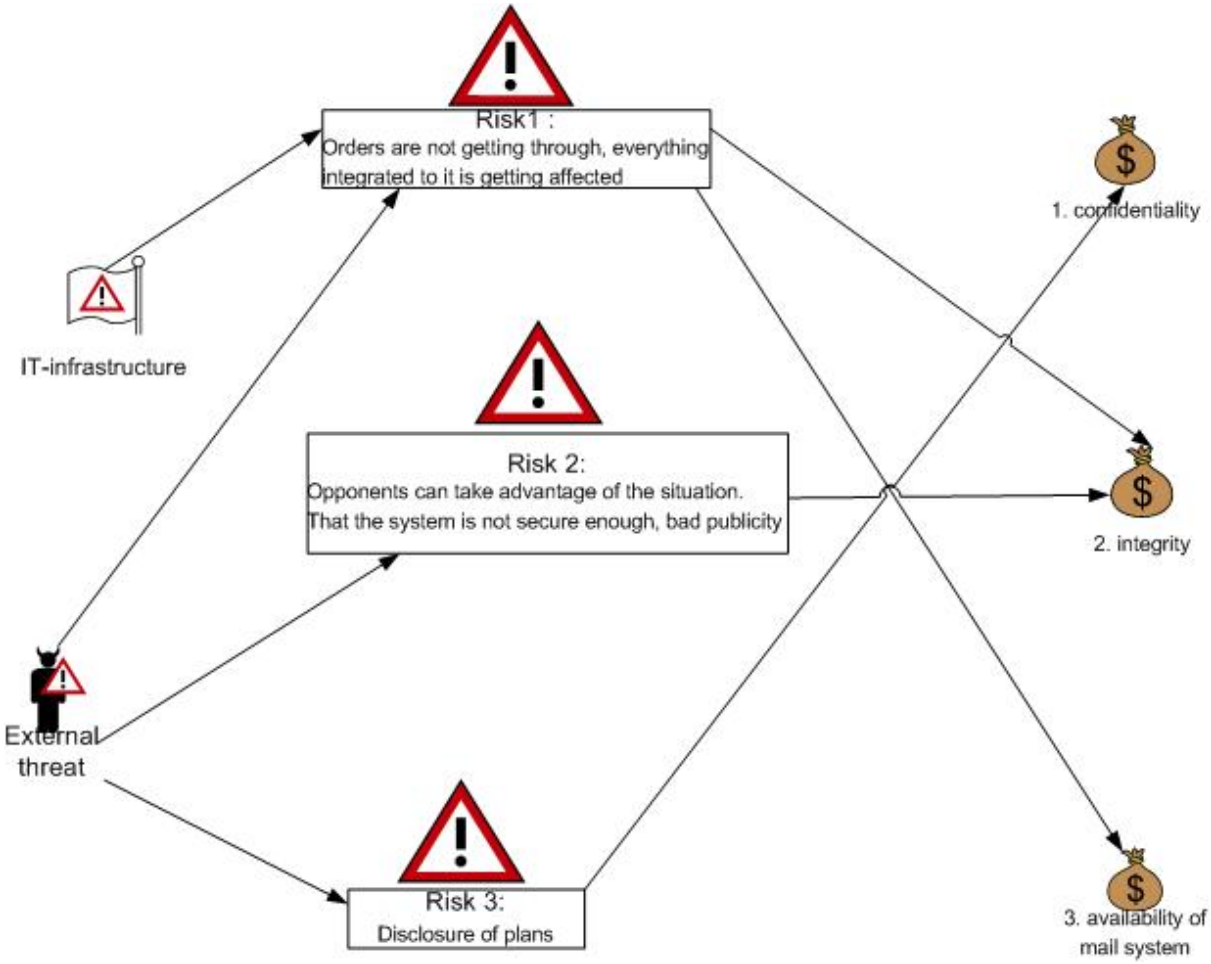


Figure 4-17: Risk diagram (case study)

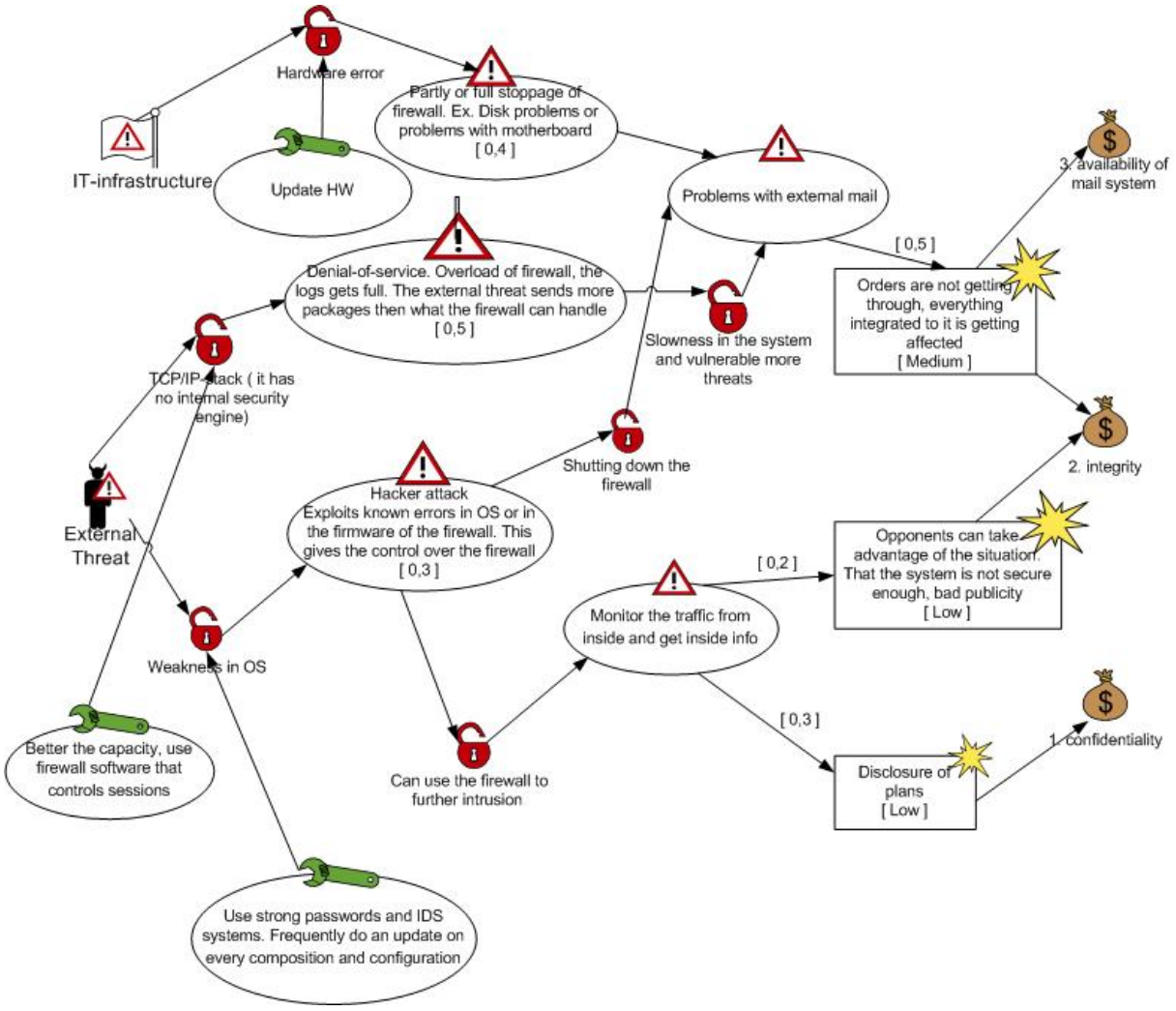


Figure 4-18: Treatment diagram (case study)

4.6 Translation of the threat diagram:

1. **IT-infrastructure** exploits vulnerability **Hardware error** to initiate **Partly or full stoppage of firewall. Ex Disk problems or problems with motherboard** with undefined likelihood.

2. **External threat** exploits vulnerability **TCP/IP-stack (it has no internal security engine)** to initiate **Denial-of-service. Overload of firewall, the logs gets full. The external threat sends more packages then what the firewall can handle** with undefined likelihood.

3. **External threat** exploits vulnerability **Weakness in OS** to initiate **Hacker attack Exploits known errors in OS or in the firmware of the firewall. This gives the control over the firewall** with undefined likelihood.

4. **Partly or full stoppage of firewall. Ex Disk problems or problems with motherboard** leads to **Problems with external mail** with undefined likelihood.

5. **Denial-of-service. Overload of firewall, the logs gets full. The external threat sends more packages then what the firewall can handle** leads to **Problems with external mail** with undefined likelihood, due to vulnerability **Slowness in the system and vulnerable to more threats**.

6. **Hacker attack. Exploits known errors in OS or in the firmware of the firewall. This gives the control over the firewall** leads to **Problems with external mail**, due to vulnerability **Shutting down the firewall**.

7. **Hacker attack. Exploits known errors in OS or in the firmware of the firewall. This gives the control over the firewall** leads to **Monitor traffic from inside and get inside info,** due to vulnerability **Can use the firewall to further intrusion**

8. **Problems with external mail** leads to **Orders are not getting, everything integrated to it is getting affected** with a likelihood of 0,5.

9. **Monitor traffic from inside and get inside info** leads to **Opponents can take advantage of the situation. That the system is not secure enough, bad publicity** with a likelihood of 0,2.

10. **Monitor traffic from inside and get inside info** leads to **Disclosure of plans** with a likelihood of 0,3.

11. **Orders are not getting, everything integrated to it is getting affected** impacts **Availability of mail system** with undefined consequence.

12. **Orders are not getting, everything integrated to it is getting affected** impacts **Integrity** with undefined consequence.

13. **Opponents can take advantage of the situation. That the system is not secure enough, bad publicity** impacts **Integrity** with undefined likelihood.

14. **Disclosure of plans** impacts **Confidentiality** with undefined consequence.

5. Method

When using CORAS diagrams in risk analysis there is no standard calculation method integrated today. It is up to every risk analyse team to decide how they want to calculate the likelihood, if it is necessary.

There is significance of one thing when calculating the probability in CORAS diagrams, it is that the likelihood for the different scenarios must be given. Then this likelihood can be used to calculate the probability for the whole scenario, or from one state to another. The state diagram gives a finer picture of how Markov chains will work. Markov chains then take this likelihood given in those different states and plots them in a transition matrix. This transition matrix uses an equation to calculate the probability for the chosen scenario.

There are some complexities in this equation when the quantity of states and scenarios gets greater, and sum of steps that must be used gets greater too. We will then be using a great time to solving the equations. We know that if we are to only use the given path's in the threat diagram, we will only multiply the different path's likelihood. But if we simulate the system according to Markov chains we get many other paths and their likelihood, which is not a part of the original threat diagram. We can use them to calculate the probability by applying Markov chains to it.

5.1 Simulation

We apply this approach of simulating the system as the different scenarios are shown in the threat diagram of the case study. We will then apply Markov chains to it in order to calculate the probability.

We will first transform the given threat diagram in the case study into a state diagram.

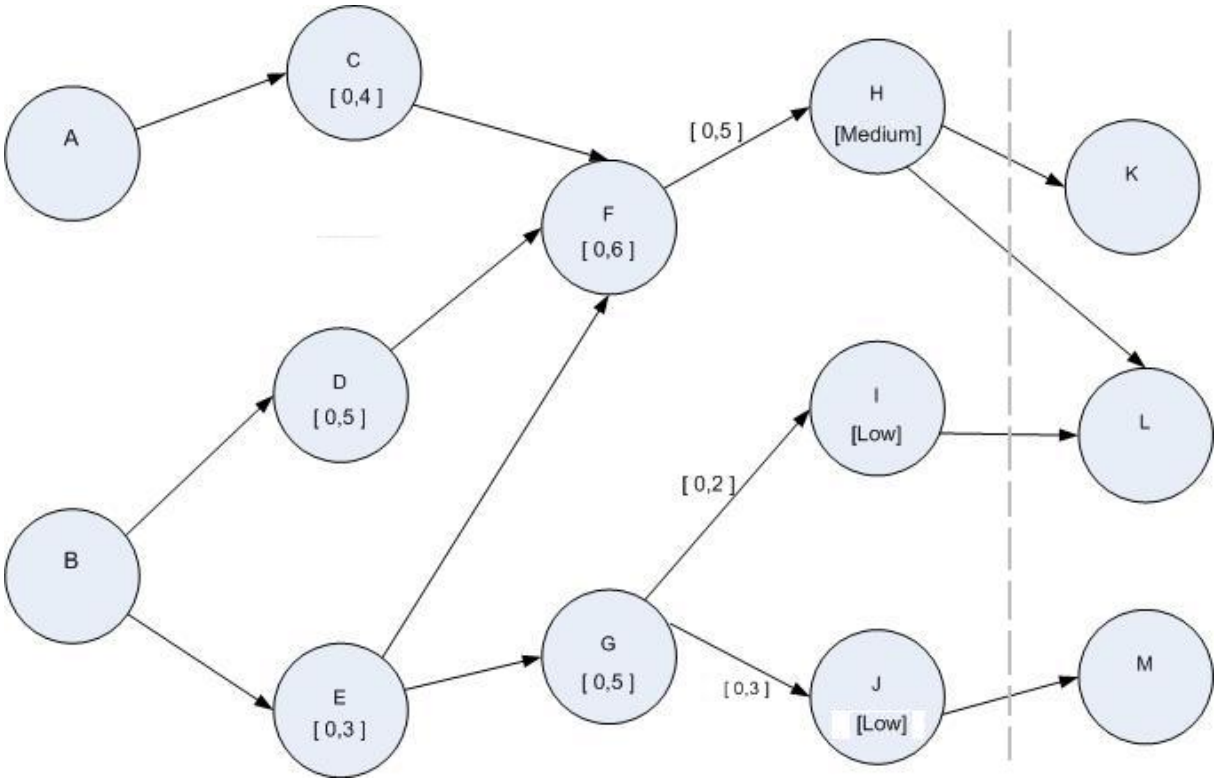


Figure 5-1: State diagram which is made by transforming the threat diagram (figure 4-16)

We have drawn a grey line to illustrate that the last three states, K L M, will not be used in the calculation of probability. This is because these three states represent the assets, and we measure the relationship between them and the previous states in consequences and not in likelihood. We do not exclude them from the state diagram, because they do give a picture of which state does affect the asset and with what probability after we have calculated it.

We now transform the state diagram into a new state diagram with those paths that are needed for applying Markov chains. The likelihoods for these paths will be calculated through simulation. Before we transform into what is needed for the simulation we have to look at some aspects of the threat diagram. We can see that there are two different threats at the starting point; those threats initiate different threat scenarios with probably different likelihoods, which then lead to either same unwanted incident, and one of threats also leads to two other unwanted incidents. This can also be seen in our state diagram. Since there are two different start states, and even if they lead to the same end state, it gives an indication that we have to treat them separately when applying Markov chains to it.

In Markov chains it is not possible to calculate probability for sequences that have different start states. We now have to segregate the state diagram in different sub state diagrams so that we can simulate correctly. We have to do this with having the original threat diagram in our mind, to avoid making new sequences which are not a part of the original one.

When we simulate we have to treat every sequence separately. What we mean here is we can see state B pointing at state D and E, state E point further on to state F and state G. We have to divide them into different sections so that the states B → D → F → H is a separate sequence, and sates B → E → F → H are separate. Every time a state point at two different states, we treat them separately. This is done because when we simulate we have to see how the system reacts to those different states. When more then one state points to the same state, or when one state points to two different states we have to treat them as different sequences.

The different sub state diagrams looks like this:

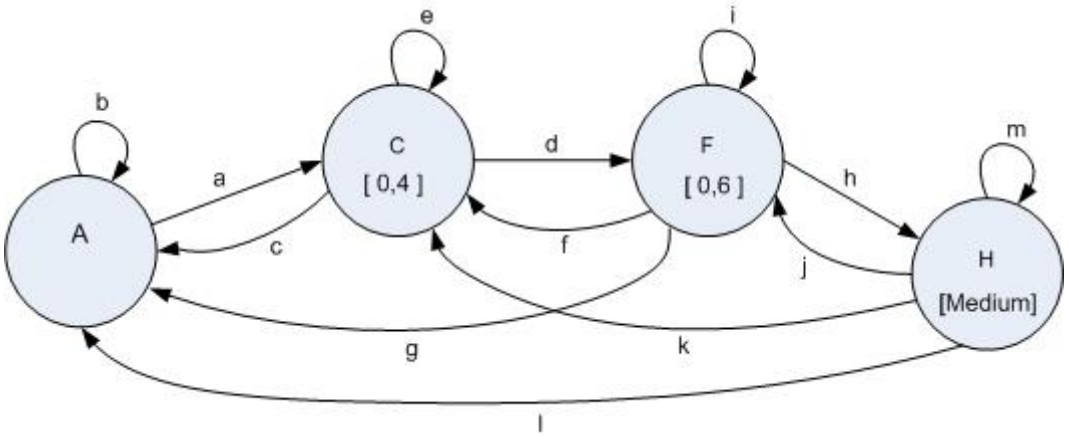


Figure 5-2: First sub state diagram

The likelihood between state F and state H is given as 0,6 but as we see in this state diagram it is not given here. The reason for this is to calculate this also by simulation and then se if we get a value around what is given in the threat diagram. Then we can see how accurate our simulation is by comparing it with the assumption by brainstorming and historical data. This comparison makes our simulation more reliable.

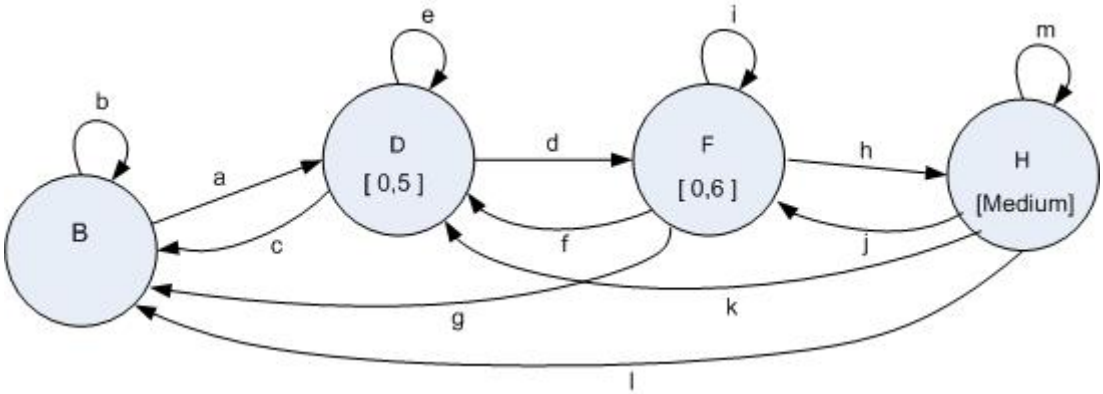


Figure 5-3: Second sub state diagram

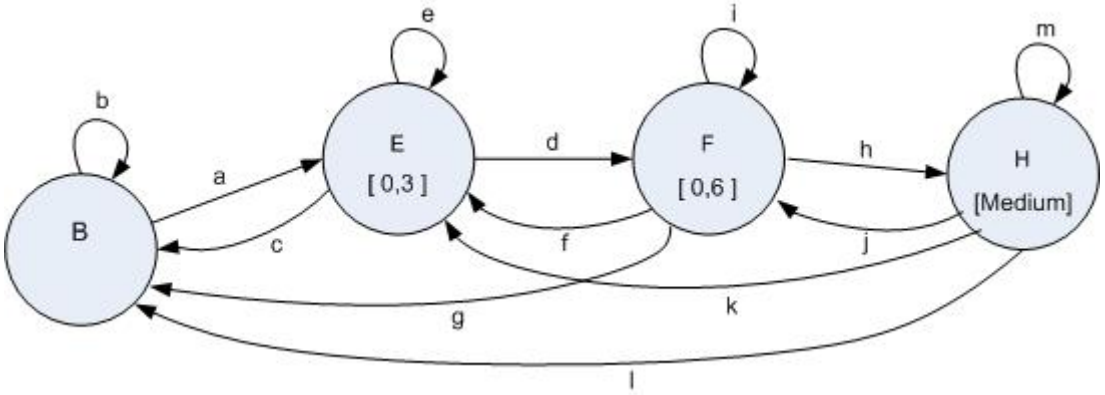


Figure 5-4: Third sub state diagram

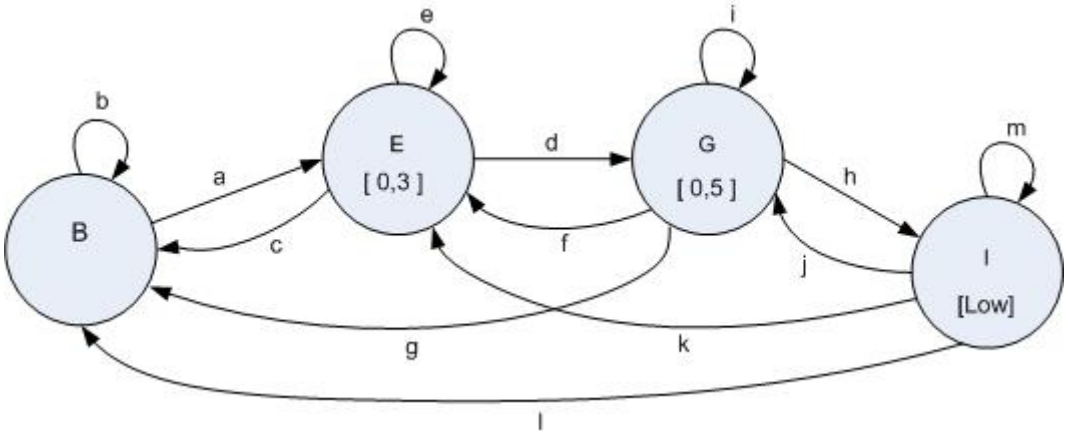


Figure 5-5: Fourth sub state diagram

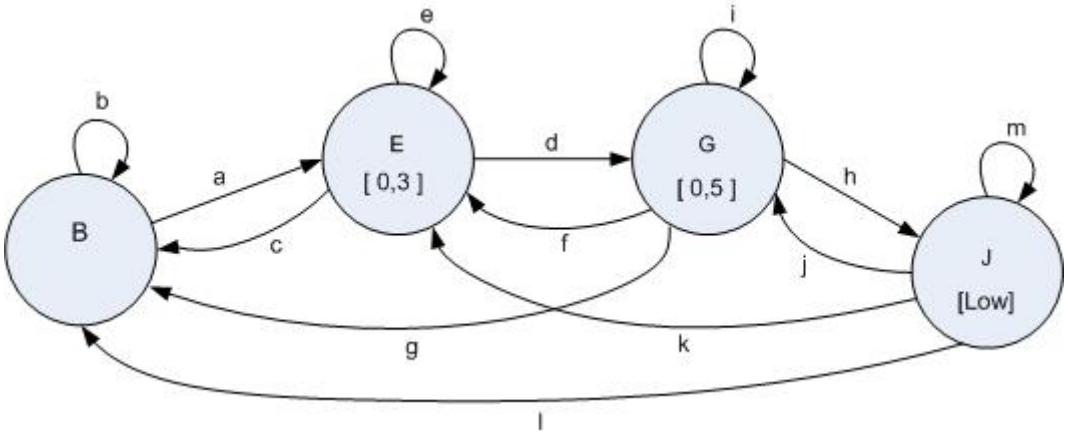


Figure 5-6: Fifth sub state diagram

These separations show how Markov chains work, how the system has to be divided if we want use Markov chains to it, and when we want to simulate it with respect to both CORAS threat diagrams and Markov chains. Each sub state diagram is looked individually and they do not influence each other at any time. If we see that two sub state diagrams uses the same states and their assumption are also be the same, then it is not necessary to simulate them

again. We can reuse the likelihood found in the previous sub state diagram. Even here we look at them separately, but use their result because they look a like and their assumptions are the same.

The rule we apply for the assumptions when simulating is:

If we give an assumption for finding the likelihood **a**, then the assumption for finding **c** can not be greater then **a**. The assumption for **d** can not be greater then **a**, and the sum of assumptions for finding **f** and **g** can not be greater then the assumption of **d**. The assumption for **h** can not be greater then the assumption for **d**, while the sum of assumptions for **j, k, l** can not be greater then **h**.

$$\textit{Assumption c} \leq \textit{Assumption a}$$

$$\textit{Assumption d} \leq \textit{Assumption a}$$

$$\textit{Sum of assumptions (f + g)} \leq \textit{Assumption d}$$

$$\textit{Assumption h} \leq \textit{Assumption d}$$

$$\textit{Sum of assumptions (j + k + l)} \leq \textit{Assumption h}$$

The likelihood for each path is found through simulation by applying the rules given above and those given in chapter 4 to calculate them. It is as mentioned in chapter 4 important that we simulate with an exact assumption. The likelihood for the different paths given in these sub state diagrams must be calculated; if they are not then it is not possible to apply Markov chains as intended. We will always make new assumption for the direct path line and for the reverse paths.

We will use our simulation tool on these sub state diagrams. It works in this manner that it takes the assumptions for those paths that are needed and does a random check. If an assumption is that it is possible to go from state 1 to state 2 five times a day, then the simulation tool tries this five times a day. We also have to put in maximum value of time for how many days, hours, months or years we want to simulate. Then this is simulated for this period of time. The answer we get will then be inserted in the probability matrix of Markov chains, and in the end use the equation to find the probability. We will here again see that if we chose the minimum amount of steps, that are those taken from the CORAS threat diagram, we will only multiply those likelihoods in the path between all the states. This simple multiplication does not use the full advantage of Markov chains. It is when we exceed the minimum of amount of steps we can see the potential of Markov chains. It is up to each analyse team to decide number of steps they want to use when applying Markov chains, but the least amount of steps is that taken from the threat diagram.

We start the simulation with the first sub state diagram. To explain how the simulation is done, and the assumption made for the simulation we will go step wise through each sub state diagram.

To give an understandable picture we divide the sub state diagrams into sections, and simulate each section step wise. In this simulation we have decided that the time period of the simulation is *20 years*.

First sub state diagram

Section one:

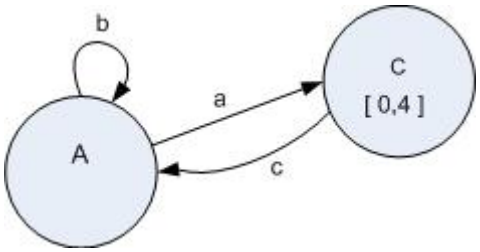


Figure 5-7: Section one of the first sub state diagram

State A: IT-infrastructure
State C: Partly or full stoppage of the firewall, ex. Disk problems or mother board

a:

- *Explanation*: The likelihood for going from state A to state C, in other words an occurrences of a problem

- *Assumption*: The firewall has 2 RAID disks but not of the best quality. It is possible that one or both of them will fail and this will include a partly or full stop of the firewall, this can occur maximum **8 times in a year**.

c:

- *Explanation*: We went from state A to state C, but because of a backup mechanism we were thrown out and sent back to state A.

- *Assumption*: The firewall computer is in a cluster. With a primary and secondary system. So when we for example get a total disk crash on the primary system, the backup system will replace it. Since the backup system will only kick in when the primary firewall system fails, the maximum number of times the backup system will kick in cannot exceed more then the primary system can fail. In other words **c** cannot be larger then **a**. But when the secondary system also fails for example in a core software problem then the primary and secondary systems will fail so we go to state C. Therefore in a good system likelihood **c** should be as high as likelihood **a**, but in our case we have poor maintenance on the infrastructure and it is not for sure that the backup system will always kick in on time, so we assume the maximum number of times the backup system will work is **3 times**

b:

- *Explanation*: The likelihood for not going from state A to state C. This can for example be that we have a power failure, and the primary system is affected, but since the company has a backup generator which kicks in or UPS, we still have a failure of the infrastructure, but it's working on a backup system.

- *Calculation*: **b = 1- a**.

Section two:

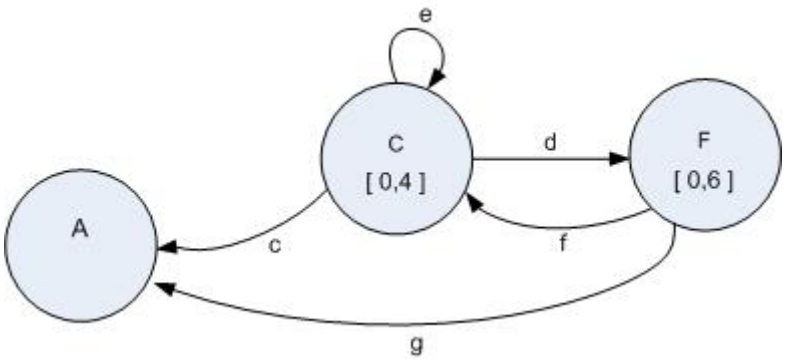


Figure 5-8: Section two of the first sub state diagram

State C: Partly of full stoppage of firewall, ex. Disk problems or mother board
State F: Problems with mail

d:

- *Explanation*: The likelihood for going from state C to state F.

- *Assumption*: There has been a total failure of the firewall, both primary and secondary systems are down. There is now a problem with the external mail. This cannot occur more times then the firewall is down, so **d** cannot be higher then **a**.
  Note. We assume that the mail will only fail when the firewalls will fail. This can occur maximum **7 times**.

f and g:

- *Explanation for f*: We went from state C to state F, but because of some sort of security mechanism we were thrown out and sent back to state C.

- *Explanation for g*: The security mechanism sent us all the way back to state A from state F.

- *Assumption*: When there has been a total failure of the firewalls, we will have problem with mail. Normally the firewall functions as a front-end server and it scans all incoming and outgoing e-mail for virus and spam, in addition to a DNS server. Because the way the infrastructure is configured, we have these scenarios.

  - Scenario 1: Total mail server crash.

  - Scenario 2: Incoming mail will function (**g**).

  - Scenario 3: Outgoing mail will be pending and waiting for the firewall (**f**)

  g and **f** cannot be higher then **d**. We assume that **g is maximum 1 times** and **f is maximum 2 times**

e:

- *Explanation*: The likelihood for not going from state C to state F, and not going back from state C to state A. There can be many reasons for this not to occur.

- *Calculation*: **e = 1- d - c**.

Section three:

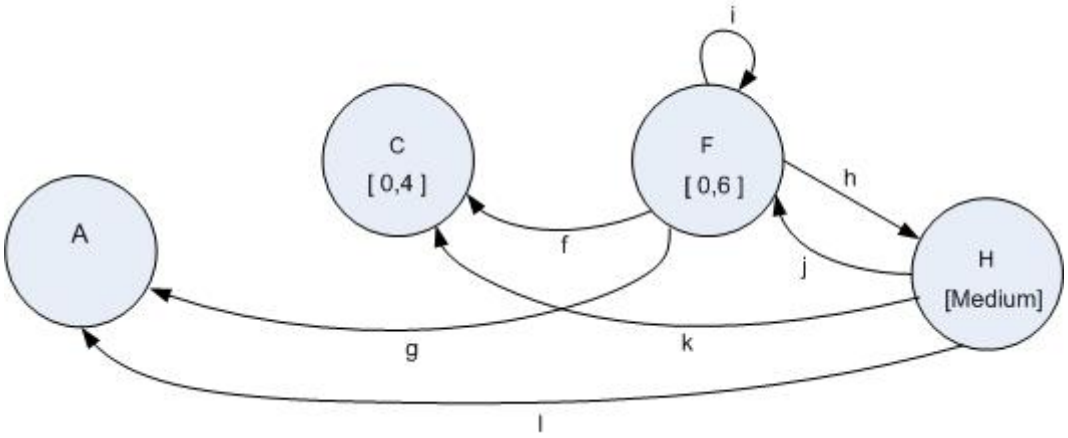


Figure 5-9: Section three of the first sub state diagram

State F: Problems with mail
State H: Orders are not getting through, everything integrated to is affected

h:

- *Explanation*: The likelihood for going from state F to state H.

- *Assumption*: Mail problems leads to this and this may occur **maximum of 6 times**.

j:

- *Explanation:* We went from state F to state H, but some security mechanism sent us back to state F.

- *Assumption*: In this scenario our CRM system is queuing the mails and waiting for it to start again. This can occur maximum **2 times**.

i:

- *Explanation*: The likelihood for not going from state F to state H, and not going back from state F to state C and state A.

- *Calculation*: **i = 1- h – f - g**

k:

- *Explanation*: We went from state F to state H, but because of some sort of security mechanism we were thrown out and sent back to state C.

- *Assumption*: We assume that the CRM system has to get external data, and since the firewall is down, we do not have the correct tcp/ip connections, and therefore the CRM system will wait for the firewall to go up. This can be maximum **1 times**.

l:

- Explanation: The security mechanism sent us all the way back to state A from state H.

- *Assumption*: We use fax machines or other equipments to inform new clients and supplier about the problem and try to collect the orders manually. The orders will then later be registered in the CRM-system and the bill will be sent after this. This is not the ideal situation for the firm, but this is done for important customers and orders. The firm will deploy this solution **maximum 1 times**.

Section four:

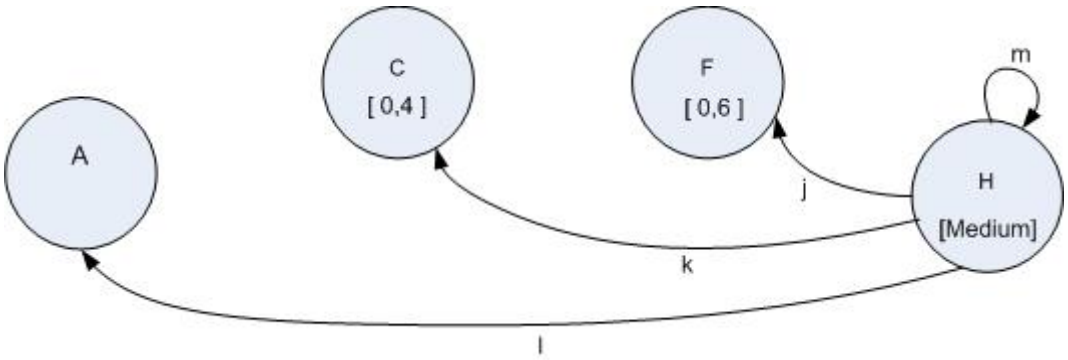


Figure 5-10: Section four of the first sub state diagram

State H: Orders are not getting through, everything integrated to is affected

m:

- *Explanation*: The likelihood for staying at state H. We see that state H is the last state, and it is only possible to go back.

- *Calculation*: **m = 1 – j – k – l**

Second sub state diagram

First section:

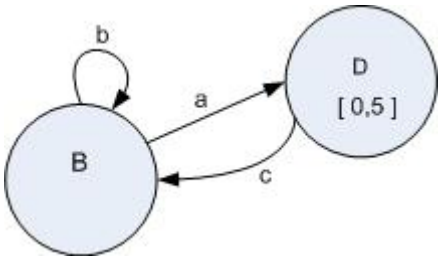


Figure 5-11: Section one of the second sub state diagram

State B: External Threat
State D: Denial-of-service

a:

- *Explanation*: The likelihood for going from state B to state D.

- *Assumption*: A hacker can use attack the system using a Denial of Service (DoS) method and overload the firewall. Since Post-Service is not a large company, we assume that the maximum of occurrences of this is **10 per year**.

c:

- *Explanation*: We went from state B to state D, but because of some sort of security mechanism we were thrown out and sent back to state B.

- *Assumption*: We have a firewall that has some protection against Denial-of-Service attack, it has the ability to stop request from single ip addresses. But when there are more then 20 sources of ip addresses the firewall cannot distinguish between large load and a Denial-of-Service attack. We therefore assume that our firewall can maximum do this **4 times**.

b:

- *Explanation*: The likelihood for not going from state A to state C, because it was not possible.

- *Calculation*: **b = 1 - a**

Section two:

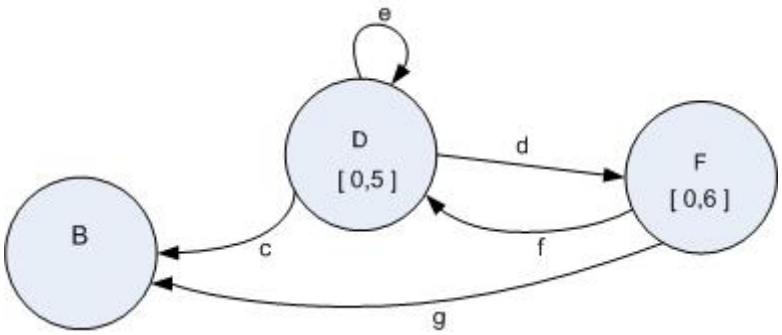


Figure 5-12: Section two of the second sub state diagram

State D: Denial-of-service
State F: Problems with mail

d:

- *Explanation*: The likelihood for going from state D to state F.

- *Assumption*: Since the hacker wants to use the mail server as a host for sending spam, the SMTP part of the mail server is affected, and outgoing mail will not work. This is done to the exploit of the firewall. This can maximum of **8 times**.

f and g:

- *Explanation for f*: We went from state D to state F, but because of some sort of security mechanism we were thrown out and sent back to state D.

- *Explanation for g*: The security mechanism sent us all the way back to state B from state F.

- *Assumption*: When there has been an attack and the firewall is breached, we will have problem with mail. Since there is an attack to exploit the mail server. We have these scenarios.

    - Scenario 1: Total mail server crash.

    - Scenario 2: Incoming mail will function (**g**).

    - Scenario 3: Outgoing mails will not work completely because of the load on the server (**f**)

  The mail server will either stop up or not function in the right manner. But the incoming mail will function. This is the likelihood **g**. The maximum number for **g** to occur is **2 times**.

  The outgoing mail will go through sometimes and sometime not, depending on the slowness of the system. This is the likelihood of **f**, and it can occur maximum of **3 times**

e:

- *Explanation*: The likelihood for not going from state D to state F, and going not back from state D to state B. There can be many reasons for this not to occur.

- *Calculation*: **e = 1 - d - c**.

Section three:

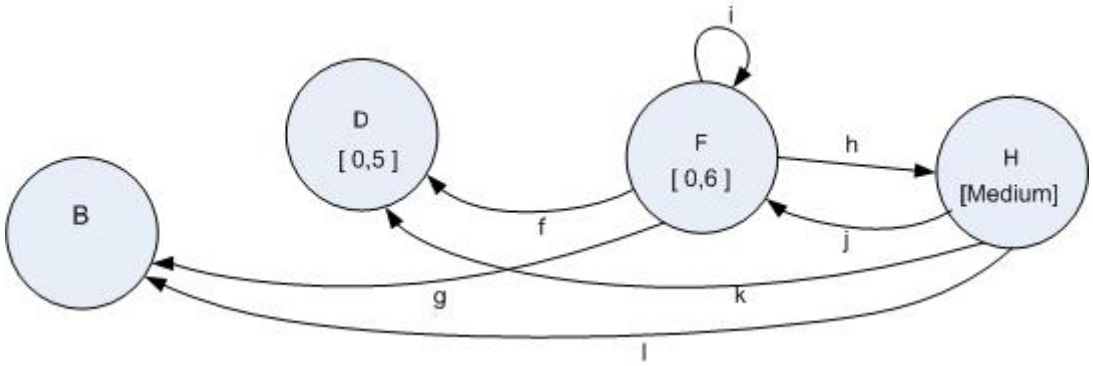


Figure 5-13: Section three of the second sub state diagram

State F: Problems with mail
State H: Orders are not getting through, everything integrated to is affected

h:

- *Explanation*: The likelihood for going from state F to state H.

- *Assumption*: Mail problems lead to this and this can occur maximum of **6 times**.

j:

- Explanation: We went from state F to state H, but some security mechanism sent us back to state F.

- *Assumption*: In this scenario our CRM system is queuing the mails and waiting for it to start again. This can occur maximum **2 times**.

i:

- *Explanation*: The likelihood for not going from state F to state H, and not going back from state F to state D, state A and state E.

- *Calculation*: **i = 1- h – f - g**

k:

- *Explanation*: We went from state F to state H, but because of some sort of security mechanism we were thrown out and sent back to state D.

- *Assumption*: Because of the Denial-of-Service attack some processes of the CRM system for validation of data will fail. This can occur maximum of **2 times**.

l:

- Explanation: The security mechanism sent us all the way back to state B from state H.

- Assumption: We use fax machines or other equipments to inform new clients and supplier about the problem and try to collect the orders manually. The orders will then later be registered in the CRM-system and the bill will be sent after this. This is not the ideal situation for the firm, but this is done for important customers and orders. The firm will deploy this solution **maximum 1 times**.

Section four:

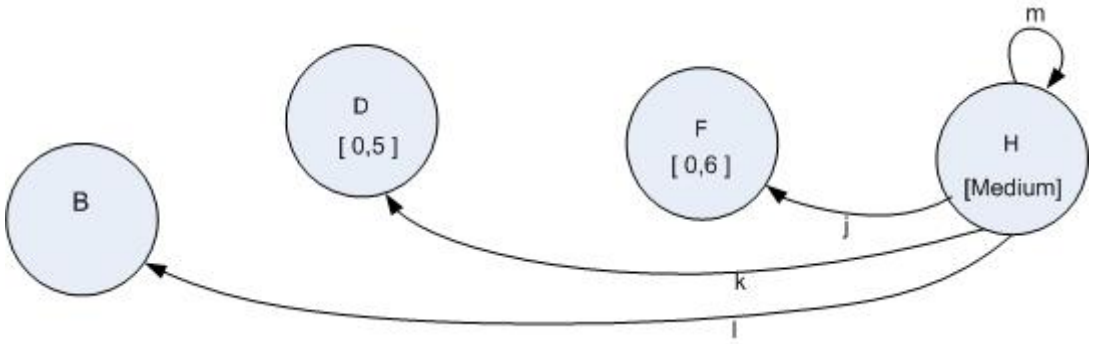


Figure 5-14: Section four of the second sub state diagram

State H: Orders are not getting through, everything integrated to is affected

m:

- *Explanation*: The likelihood for staying at state H. We see that state H is the last state, and it is only possible to go back.

- *Calculation*: **m = 1 – j – k – l**

Third sub state diagram

Section one:

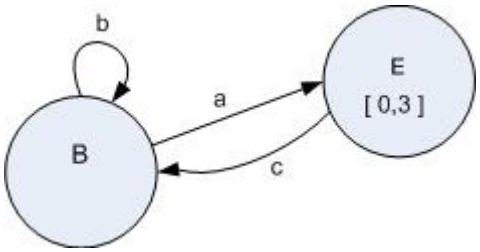


Figure 5-15: Section one of the third sub state diagram

State B: External threat
State E: Hacker attack

a:

- *Explanation*: The likelihood for going from state B to state E.

- *Assumption*: External threat exploits a weakness in the OS and tries to take control over the firewall. This is possible maximum **8 times** per year.

c:

- *Explanation*: We went from state B to state E, but because of some sort of security mechanism we were thrown out and sent back to state B.

- *Assumption*: There is a possibility that the hacker has not the ability to fully exploit the OS problem, and therefore is kicked out of the system. Or the OS gets a blue screen due to the possible exploit and reboots, the hacker is then again kicked out of the system, maximum **3 times**.

b:

- *Explanation*: The likelihood for not going from state B to state E, because it was not possible.

- *Calculation*: **b = 1 - a**

Section two:

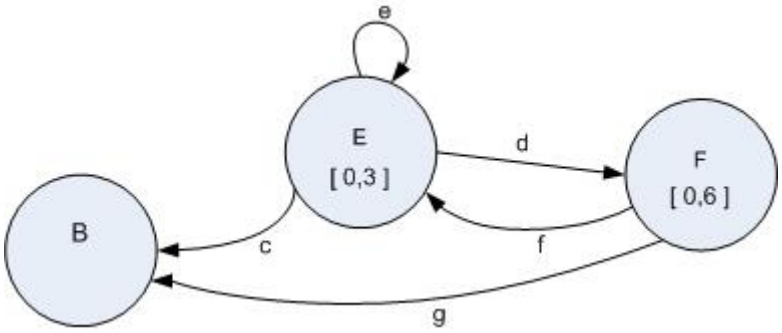


Figure 5-16: Section two of the third sub state diagram

State E: Hacker attack
State F: Problems with mail

d:

- *Explanation*: The likelihood for going from state E to state F.

- *Assumption*: The hacker attack is shutting down the firewall which then creates mail problems. The mail problems maximum to occur is **7 times**.

f and g:

- *Explanation for f*: We went from state E to state F, but because of some sort of security mechanism we were thrown out and sent back to state E.

- *Explanation for g*: The security mechanism sent us all the way back to state B from state F.

- *Assumption*: When there has been a total failure of the firewall, we will be having problems with mail. Normally the firewall functions as a front-end server and it scans all incoming and outgoing e-mail for virus and spam, in addition to a DNS server. We have these scenarios because the way the infrastructure is configured.

    - Scenario 1: Total mail server crash.

    - Scenario 2: Incoming mail will function (**g**).

    - Scenario 3: Outgoing mail will be pending and waiting for the firewall (**f**)

  g and **f** cannot be higher then **d**. We assume that **g is 2 times** and **f is 3 times**

e:

- *Explanation*: The likelihood for not going from state E to state F, and not going back from state E to state B.

- *Calculation*: **e = 1 - d – c**

Section three:

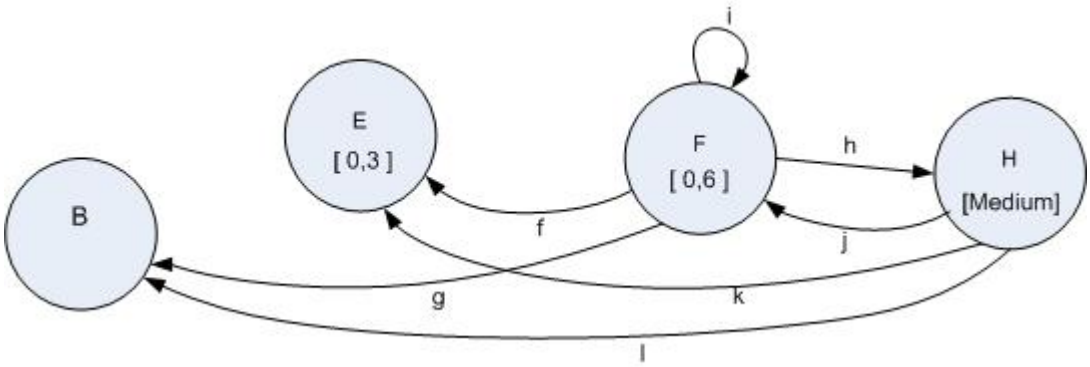


Figure 5-17: Section three of the third sub state diagram

h:

- *Explanation*: The likelihood for going from state F to state H.

- *Assumption*: Mail problems lead to this and this can occur maximum of **6 times**.

j:

- Explanation: We went from state F to state H, but some security mechanism sent us back to state F.

- *Assumption*: In this scenario our CRM system is queuing the mails and waiting for it to start again. This can occur maximum **2 times**.

i:

- *Explanation*: The likelihood for not going from state F to state H, and not going back from state F to state D, state A and state E.

- *Calculation*: **i = 1- h – f - g**

k:

- *Explanation*: We went from state F to state H, but because of some sort of security mechanism we were thrown out and sent back to state E.

- *Assumption*: We assume that the CRM system has to get external data, and since the firewall is down, we do not have the correct tcp/ip connections, and therefore the CRM system will wait for the firewall to go up. This can occur maximum **1 times**.

l:

- Explanation: The security mechanism sent us all the way back to state B from state H.

- Assumption: We use fax machines or other equipments to inform new clients and supplier about the problem and try to collect the orders manually. The orders will then later be registered in the CRM-system and the bill will be sent after this. This is not the ideal situation for the firm, but this is done for important customers and orders. The firm will deploy this solution **maximum 1 times**.

Section four:

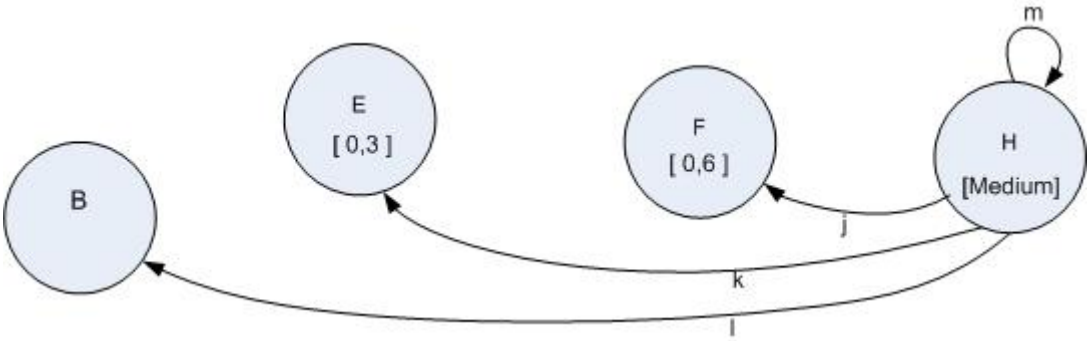


Figure 5-18: Section four of the third sub state diagram

State H: Orders are not getting through, everything integrated to is affected

m:

- *Explanation*: The likelihood for staying at state H. We see that state H is the last state, and it is only possible to go back.

- *Calculation*: **m = 1 – j – k – l**

Fourth sub state diagram

First section:

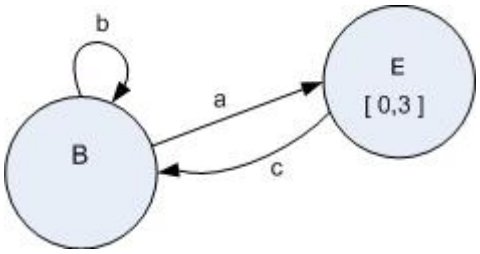


Figure 5-19: Section one of the fourth sub state diagram

The likelihood for **a**, **c, and b** is the same as in the third sub state diagram.

Section two:

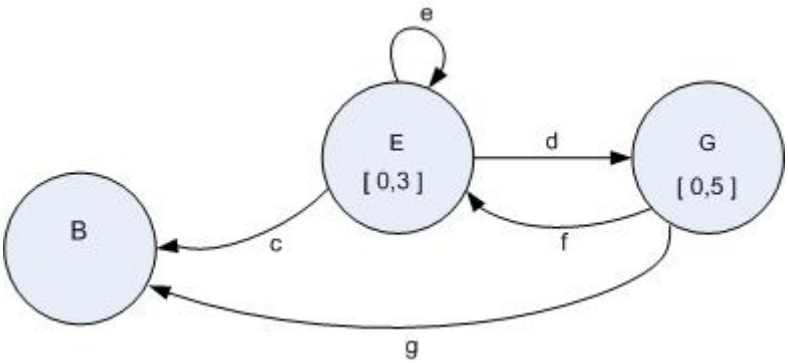


Figure 5-20: Section two of the fourth sub state diagram

State E: Hacker attack
State G: Monitoring the traffic from inside and getting inside information

d:

- *Explanation*: The likelihood for going from state E to state G.

- *Assumption*: The firewall can be used for further intrusion. This can occur maximum of **5 times**.

f and g:

- *Explanation f*: We went from state E to state G, but because of some sort of security mechanism we were thrown out and sent back to state E.

- *Explanation g:* The security mechanism sent us all the way back to state B from state G.

- *Assumption*: Since we have an infrastructure that does not allow somebody to log in to the system from the internet for a long period of time. If the hackers is in the system for more then 5 minutes he is automatically kicked out. If the hacker is not able to shut down this security mechanism he is kicked out of the system (**g**). The hacker has been able to shut down this security feature, but has not been able to install any software to monitor the traffic (**f**). The maximum of occurrences for **f is 2 times** and for **g is 2 times**.

e:

- *Explanation*: The likelihood for not going from state E to state G, and not going back from state E to state B. There can be many reasons for this not to occur.

- *Calculation*: **e = 1- d - c**.

Section three:

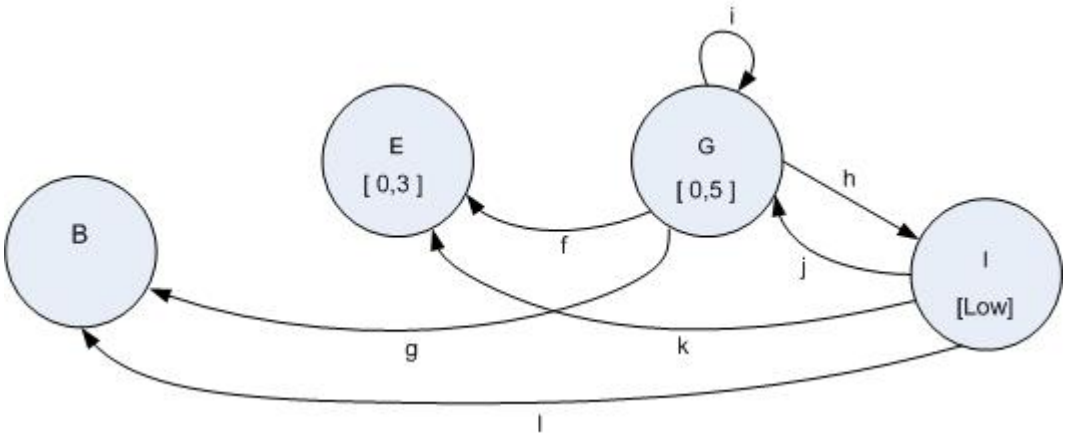


Figure 5-21: Section three of the fourth sub state diagram

State G: Monitoring the traffic from inside and therefore getting inside information
State I: Opponents can take advantage of the situations, bad publicity etc

h and j:

- *Explanation h*: The likelihood for going from state G to state I.

- *Explanation j:* We went from state G to state I, but some security mechanism sent us back to state G.

- The hacker is inside the system and monitoring the traffic, but there are not so many occasions where information sent between the systems that the hacker can use for bad publicity. The number of times he is successful in getting information is (**h**) and occurs **maximum of 3 times**, but much of the information is not useful and he/she is only successful **maximum 2 time (j).**

i:

- *Explanation*: The likelihood for not going from state G to state I, and going back from state G to state E and state B.

- *Calculation*: **i = 1 - h - f - g**

k:

- *Explanation*: We went from state G to state I, but because of some sort of security mechanism we were thrown out and sent back to state E.

- *Assumption*: If the user is success in shutting down the 5 minute limit for external connections and has collected some internal information, then we do not have any security mechanism to kick out the hacker from the system. This can not occur in our system, so we can not simulate this path. We will add **0** as the likelihood for this to occur in our calculation later.

l:

- Explanation: The security mechanism sent us all the way back to state B from state I.

- *Assumption*: Same assumption as **k**.

Section four:

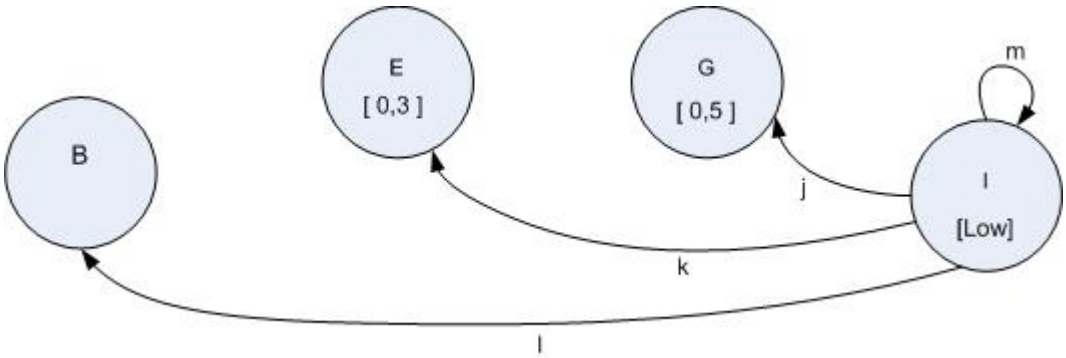


Figure 5-22: Section four of the fourth sub state diagram

State I: Opponents can take advantage of the situations, bad publicity etc.

m:

- *Explanation*: The likelihood for staying at state I. We see that state I is the last state, and it is only possible to go back.

- *Calculation*: **m = 1 – j – k – l**

Fifth sub state diagram

First and second section is the same as in the third sub state diagram. So we will simulate from *section three:*

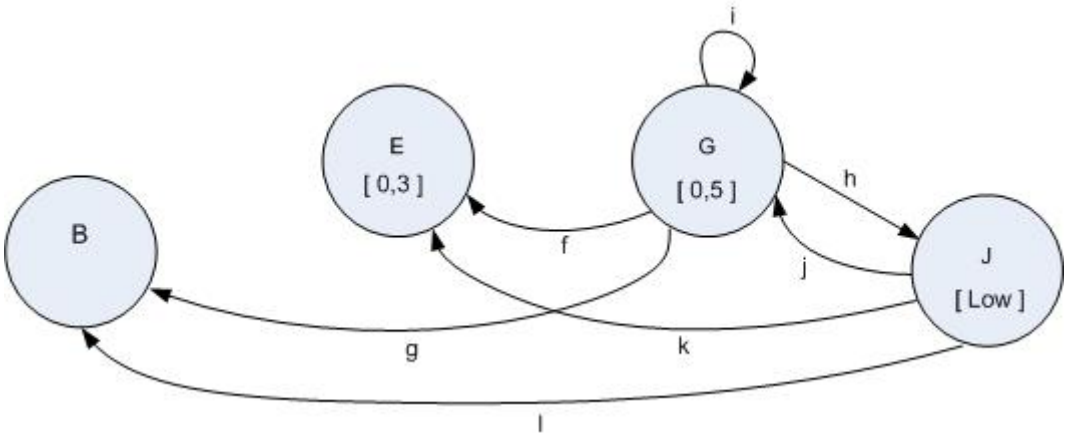


Figure 5-23: Section three of the fifth sub state diagram

State G: Monitoring the traffic from inside and getting inside information
State J: Disclosure of plans

h and j:

- *Explanation*: The likelihood for going from state G to state J.

- The hacker is inside the system and monitoring the traffic, but there are some occasions where there are sent sensitive information between users that the hacker can use for disclosing of plans. The number of times he is successful in getting information is (**h**) and occurs **maximum of 4 times**. Some times the information is not useful and he/she is only successful **maximum 1 time (j)**.

i:

- *Explanation*: The likelihood for not going from state G to state J, and going back from state G to state E and state B.

- *Calculation*: **i = 1- h – f - g**

k:

- *Explanation*: We went from state G to state J, but because of some sort of security mechanism we were thrown out and sent back to state E.

- *Assumption*: If the user is success in shutting down the 5 minute limit for external connections and has collected some internal information, then we do not have any security mechanism to kick out the hacker from the system. This can not occur in our system, so we can not simulate this path. We will add **0** as the likelihood for this to occur in our calculation later.

l:

- *Explanation:* The security mechanism sent us all the way back to state B from state J.

- *Assumption:* The same assumption as **k**.

Section four:

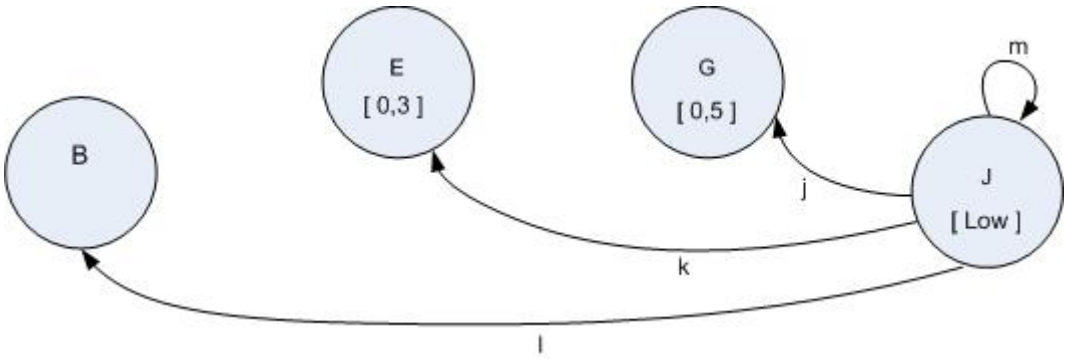


Figure 5-24: Section four of the fifth sub state diagram

State J: Disclosure of plans

m:

- *Explanation*: The likelihood for staying at state J. We see that state J is the last state, and it is only possible to go back.

- *Calculation*: **m = 1 – j – k – l**

5.2 Applying Markov chains to calculate the probability

We have simulated and calculated the likelihood for each section for all sub state diagrams. We know have to apply Markov chains to calculate the probability. In this thesis we are going to calculate the probability for the first sub state diagram in this thesis to show how this works. The rest of the sub state diagrams can also be simulated and calculated, but we have chosen not to show this. The reason beyond this is to give the readers the opportunity to test our simulation tool. When simulating, the reader can use the user manual and make use of the assumptions made earlier for the rest of the sub state machines.

We will calculate the probability by going from state A → state H. We will be using three different amounts of steps, this to show that the probability changes when we apply more and more steps. The lowest amounts of steps are those taken from the CORAS threat diagram (figure 4-16), which are 3-steps. Then we will apply 5-steps, and in the last 6-steps. The more steps we apply the more use of the absorbing and reverse likelihood we do. We will then see what the probability is when only using those likelihoods that are from paths that are included in the CORAS threat diagram, and how the probability differs when we use more steps.

When making the transition matrix for Markov chains, we use Chapman-Kolmogorov equation [2].

$$P^{(n)} = P^{(n-s)} P^{(s)}$$

Because $P^{(1)} = P$ follows from the Chapman-Kolmogorov equation that $P^{(2)} = P^2$ and in general, $P^{(n)} = P^n$. The n-step transition matrix $P^{(n)}$ is just the nth power of P. The elements of P^n are the n-step transition probabilities, $P^{(n)}_{ij}$.

We have used the program Maple 11.0 [16] when making the transition matrix. Here we can make a transition matrix, insert the values, set amount of steps to be used, and it will calculate for all possible sequences. We will use this program to find all the possibilities for a given sequence, state A → state H, with a specified amount of steps. We will not insert the calculated values from the simulation, but we will make the transition matrix using the names of the paths, ex. a, b, c, etc. This will give us an equation where we can insert the values later and calculate the probability. The equations we got from using Maple 11.0 are explained below, and we have inserted these in our simulation tool. If there are many states and we want to find the probability in different amount of steps using Markov chains, it is easier to use a Math program as Maple 11.0 to find the exact sequence of the equation.

It is important to remember that if the reader wants to find the probability for some other scenarios, for example state C → state H, then he/she must use Maple 11.0 to find the sequences equation. Then they can insert the calculate values from the simulation to find the probability for going from state C → state H.

We have inserted the paths equation we got from Maple 11.0 in our simulation tool, so when it calculates the probability using Markov chains it uses those equations.

In Maple 11.0 the transition matrix made is the opposite manner then we have used in our thesis. It is built in this manner:

$$\begin{array}{c} \diagdown \\ i \end{array}\!\!\begin{array}{c} j \\ \end{array} \begin{array}{cc} 1 & 2 \end{array}$$
$$\begin{array}{c} 1 \\ 2 \end{array} \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{bmatrix} \quad (1)$$

The explanation for this matrix in plain text is:
Matrix ({(1, 1) = m[1, 1], (1, 2) = m[1, 2], (2, 1) = m[2, 1], (2, 2) = m[2, 2]})

The matrix we have used so far in this thesis looks like this:

$$\begin{array}{c} \diagdown \\ j \end{array}\!\!\begin{array}{c} i \\ \end{array} \begin{array}{cc} 1 & 2 \end{array}$$
$$\begin{array}{c} 1 \\ 2 \end{array} \begin{bmatrix} m_{1,1} & m_{2,1} \\ m_{1,2} & m_{2,2} \end{bmatrix} \quad (2)$$

The explanation for this matrix in plain text is:
Matrix ({(1, 1) = m[1, 1], (2, 1) = m[2, 1], (1, 2) = m[1, 2], (2, 2) = m[2, 2]})

It is allowed to use both methods when making the transition matrix. Since we now will use the answers from Maple program we will be making the matrix in the same manner (like matrix 1).

To understand how we take out the selected sequence that we have decided to apply Markov chains to, we look at this example:

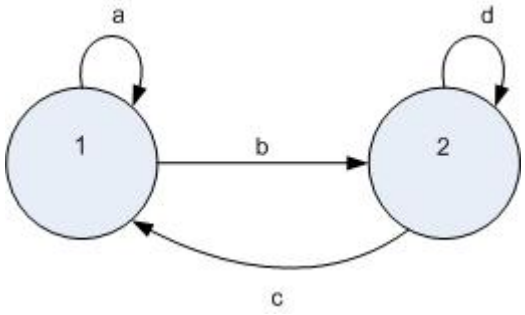


Figure 5-25: Example of a sub state diagram

We want to find the probability from state 1 → state 2, by using 2-steps.
We make a transition matrix from this diagram:

$$\begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{1} & \mathbf{2} \end{array} \\ \begin{bmatrix} a & b \\ c & d \end{bmatrix}^2 \end{array}$$

The result we get are the different sequences:

$$\begin{bmatrix} a^2 + b\,c & a\,b + b\,d \\ c\,a + d\,c & b\,c + d^2 \end{bmatrix}$$

i. The sequence for going from state 1 → state 1, in 2-steps:
[a^2 + bc]

ii. The sequence for going from state 1 → state 2, in 2-steps:
[ab + bd]

iii. The sequence for going from state 2 → state 1, in 2-steps:
[ca + dc]

iv. The sequence for going from state 2 → state 2, in 2-steps:
[bc + d^2]

We see that to calculate the probability from state 1 → state 2 we have to use the sequence given in ii.

We use this same method when finding the sequence for probability calculation from state A → state H.

The matrix without the inserting the values looks like this:

$$\begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} \end{bmatrix}$$

The explaination of this transition matrix in plain text:

Matrix({(1, 1) = m[1, 1], (1, 2) = m[1, 2], (1, 3) = m[1, 3], (1, 4) = m[1, 4], (2, 1) = m[2, 1], (2, 2) = m[2, 2], (2, 3) = m[2, 3], (2, 4) = m[2, 4], (3, 1) = m[3, 1], (3, 2) = m[3, 2], (3, 3) = m[3, 3], (3, 4) = m[3, 4], (4, 1) = m[4, 1], (4, 2) = m[4, 2], (4, 3) = m[4, 3], (4, 4) = m[4, 4]})

We insert the values:

	A	C	F	H
A	b	a	0	0
C	c	e	d	0
F	g	f	i	h
H	l	k	j	m

The textual explanation:

> *Matrix ({(A, A) = b, (A, C) = a, (A, F) = 0, (A, H) = 0, (C, A) = c, (C, C) = e, (C, F) = d, (C, H) = 0, (F, A) = g, (F, C) = f, (F, F) = i, (F, H) = h, (H, A) = l, (H, C) = k, (H, F) = j, (H, H) = m})*

We will first find the path for calculating the probability in 3-steps.

$$\begin{array}{c} \\ \mathbf{A} \\ \mathbf{C} \\ \mathbf{F} \\ \mathbf{H} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{A} & \mathbf{C} & \mathbf{F} & \mathbf{H} \end{array} \\ \begin{bmatrix} b & a & 0 & 0 \\ c & e & d & 0 \\ g & f & i & h \\ l & k & j & m \end{bmatrix}^3 \end{array}$$

The different sequences we get are:

$$\begin{aligned}
&[[(b^2+a\,c)\,b+(b\,a+a\,e)\,c+a\,d\,g,(b^2+a\,c)\,a+(b\,a+a\,e)\,e\\
&\quad +a\,d\,f,(b\,a+a\,e)\,d+a\,d\,i,a\,d\,h],\\
&\quad [(c\,b+e\,c+d\,g)\,b+(a\,c+e^2+d\,f)\,c+(e\,d+d\,i)\,g+d\,h\,l,(c\,b\\
&\quad +e\,c+d\,g)\,a+(a\,c+e^2+d\,f)\,e+(e\,d+d\,i)\,f+d\,h\,k,(a\,c+e^2\\
&\quad +d\,f)\,d+(e\,d+d\,i)\,i+d\,h\,j,(e\,d+d\,i)\,h+d\,h\,m],\\
&\quad [(g\,b+f\,c+i\,g+h\,l)\,b+(g\,a+f\,e+i\,f+h\,k)\,c+(d\,f+i^2\\
&\quad +h\,j)\,g+(i\,h+h\,m)\,l,(g\,b+f\,c+i\,g+h\,l)\,a+(g\,a+f\,e+i\,f\\
&\quad +h\,k)\,e+(d\,f+i^2+h\,j)\,f+(i\,h+h\,m)\,k,(g\,a+f\,e+i\,f+h\,k)\,d\\
&\quad +(d\,f+i^2+h\,j)\,i+(i\,h+h\,m)\,j,(d\,f+i^2+h\,j)\,h+(i\,h+h\,m)\,m\\
&\quad],\\
&\quad [(l\,b+k\,c+j\,g+m\,l)\,b+(l\,a+k\,e+j\,f+m\,k)\,c+(k\,d+j\,i\\
&\quad +m\,j)\,g+(h\,j+m^2)\,l,(l\,b+k\,c+j\,g+m\,l)\,a+(l\,a+k\,e+j\,f\\
&\quad +m\,k)\,e+(k\,d+j\,i+m\,j)\,f+(h\,j+m^2)\,k,(l\,a+k\,e+j\,f+m\,k)\,a\\
&\quad +(k\,d+j\,i+m\,j)\,i+(h\,j+m^2)\,j,(k\,d+j\,i+m\,j)\,h+(h\,j+m^2)\,m\\
&\quad]]
\end{aligned}$$

This answer is divided in this manner:

i. The sequence for going from state A → state A, in 3-steps:

[$(b^2 + ac)b + (ba + ae)c + a\,d\,g$]

ii. The sequence for going from state A → state C, in 3-steps:

[$(b^2 + ac)a + (ba + ae)e + a\,d\,f$]

iii. The sequence for going from state A → state F, in 3-steps:

[$(ba + ae)d + a\,d\,i$]

iv. The sequence for going from state A → state H, in 3-steps:

[a d h]

It is sequence **iv** we were looking for. This is inserted in our simulation tool.

The rest of the sequences are not of interest to us, but to give a clarification of what sequences they represent we can give an explanation:

[state C → state A, state C → state C, state C→ state F, state C → state H],
[state F → state A, state F → state C, state F → state F, state F → state H],
[state H → state A, state H → state C, state H → state F, state H → state H]

In the rest of the thesis we will only be showing the sequence from state A → state H we have found through using Maple program.

The transition matrix in 5-steps:

$$\begin{array}{c} \\ \mathbf{A} \\ \mathbf{C} \\ \mathbf{F} \\ \mathbf{H} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{A} & \mathbf{C} & \mathbf{F} & \mathbf{H} \end{array} \\ \begin{bmatrix} b & a & 0 & 0 \\ c & e & d & 0 \\ g & f & i & h \\ l & k & j & m \end{bmatrix}^5 \end{array}$$

The sequence from state A → state H:

$$[\,((b^2 + ac)ad + (ba + ae)(ed + di) + ad(df + i^2 + hj))h + ((ba + ae)dh + ad(ih + hm))m\,]$$

The transition matrix in 6-steps:

$$\begin{array}{c} \\ \mathbf{A} \\ \mathbf{C} \\ \mathbf{F} \\ \mathbf{H} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{A} & \mathbf{C} & \mathbf{F} & \mathbf{H} \end{array} \\ \begin{bmatrix} b & a & 0 & 0 \\ c & e & d & 0 \\ g & f & i & h \\ l & k & j & m \end{bmatrix}^6 \end{array}$$

The sequence from state A → state H:

$$[\,((b^2 + ac)b + (ba + ae)c + adg)adh + ((b^2 + ac)a + (ba + ae)e + adf)((ed + di)h + dhm) + ((ba + ae)d + adi)((df + i^2 + hj)h + (ih + hm)m) + adh((kd + ji + mj)h + (hj + m^2)m)\,]$$

5.2.1 The result

When we entered the values of our assumptions for the first sub state diagram, these are the results we got:

Attacks	1/8	1/8	8/8	2/8	1/8	8/8	1/8	3/8	0/8	0/8	7/8	5/8	0/8	4/8	6/8	0/8	6/8	0/8	6/8	7/8
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

a value : 0,4125

Attacks	1/8	0/8	0/8	3/8	1/8	0/8	2/8	0/8	2/8	2/8	2/8	3/8	3/8	0/8	2/8	2/8	3/8	3/8	3/8	1/8
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

c value : 0,20625

Attacks	5/8	0/8	3/8	6/8	5/8	4/8	1/8	2/8	2/8	2/8	3/8	0/8	7/8	0/8	5/8	7/8	7/8	1/8	2/8	2/8
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

d value : 0,4

Attacks	0/7	2/7	2/7	2/7	1/7	0/7	0/7	1/7	0/7	0/7	1/7	2/7	0/7	1/7	0/7	2/7	1/7	2/7	2/7	1/7
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

f value : 0,142857158184052

Attacks	6/7	2/7	0/7	0/7	4/7	1/7	2/7	1/7	4/7	3/7	6/7	2/7	6/7	5/7	4/7	6/7	1/7	3/7	6/7	4/7
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

h value : 0,471428537368774

Attacks	2/6	0/6	0/6	2/6	2/6	1/6	2/6	2/6	0/6	2/6	1/6	1/6	1/6	1/6	2/6	0/6	0/6	1/6	0/6	0/6
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

j value : 0,166666686534882

Attacks	0/6	1/6	1/6	1/6	1/6	0/6	0/6	0/6	1/6	1/6	1/6	1/6	0/6	1/6	1/6	1/6	0/6	1/6	0/6	0/6
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

k value : 0,099999988079071

Attacks	1/7	1/7	0/7	1/7	0/7	1/7	0/7	0/7	1/7	0/7	1/7	1/7	1/7	1/7	0/7	1/7	1/7	0/7	1/7	0/7
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

g value : 8,57142984867096E-02

Attacks	1/6	1/6	1/6	1/6	0/6	1/6	1/6	0/6	0/6	1/6	1/6	0/6	0/6	0/6	1/6	1/6	1/6	1/6	1/6	0/6
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

l value : 0,108333325386047

b : 0,5875
e : 0,39375
i ; 0,771428543329238
m : 0,791666686534882

We can see that the likelihood for ***h*** is in the region of what was given in our threat diagram. The result could have been more precise if we had the possibility to test this for an existent company. Then we could compare the simulation result with exact numbers of likelihood.

Post Service is a fictitious company, but the scenarios, likelihoods and assumptions we have made are relatively realistic. The scenarios, likelihoods and assumptions are made by brainstorming on what is actually possible in real life. This is why the answer we get from our

simulation tool can be looked as valid.

We now want to find the probability using Markov chains by using 3-steps, 5-steps and 6-steps.

In 3-steps: 7,77857086658477E-02
In 5 steps: 0,33148477819344
In 6 steps: 0,481578560773484

These answers do not only show us that the probability varies when we change the amount of steps, they also show how the probability actually changes.
This is Markov chains´ distinctive ability, and it makes our probability calculations more reliable and efficient. We can find the exact probability for a given scenario for any given amount of steps.

6. Simulation tool

We made the simulation tool in active server page (ASP) programming language, where we have simulated using the built in randomization function, to give us random number of attacks in the boundaries defined by the user. The average is then calculated and shown for each step. We use this average as the likelihood for each path and can insert it into the sub state diagram to have a better overview.

We have inserted all the rules that are needed for simulating the values for this case study. It takes the assumption inserted into it and simulates according to them. We check if any of the given assumption does breach some of our rules. If it does we give an error message telling what went wrong. It calculates the absorbing likelihoods, ***b***, ***d***, ***i***, after having simulated the first values. If we then get a negative answer in the absorbing likelihood we get an error message telling which one is of negative value. We then have to start the simulation again manually, because this indicates that something was wrong with our assumptions.

Since this is a beta version, the simulation software is customized to the case study model.

Since this is a web-application our tool can be found using this URL:
http://www.semat.no/simulering/

User manual explains how this simulation tool should be used.

The simulation tool can be developed further to a more general simulation tool, so that the number of states and paths can be made as needed. There should be done more work on the interface of the program and the way it presents the results.

The simulation tools source code can be found under Appendix B.

Recommendations for feature development:

- Make the simulation tool more general to make it able to handle more states and paths
- Make better graphic user interface

7. Discussion

We have seen how CORAS analysis is build and how it is deployed. Our main area in CORAS analysis has been the CORAS threat diagrams. We have shown through its semantics the rules for them and how they should be interpreted. This is where thought Markov chains could be suitable. The reason for this is that Markov chains are used to calculate the probability, and in CORAS analysis the main section for this is the threat diagram.

The way CORAS threat diagrams are build, its rules and regulations, it was not possible to deploy Markov chains directly with out any changes. Markov chains are used, in simple words, to find likelihood from one state to another. To do this it is essential that the likelihood is given for the different states to occur.

The second part was that to get full utilization of Markov chains then much other likelihood paths must also be known. In Markov chains it is possible for a state to point back to all of the previous states, to point back to it self, and to point to all of the next states. All of these paths can then have likelihoods, and these likelihoods can be used in the calculation of probability. These paths are not a part of the CORAS analysis, since CORAS is not a direct transcription of how the system runs, but it tells how a threat will behave and affect the system. The basis for the CORAS analysis is the brainstorming and historical data phase. The CORAS threat diagram tells where the vulnerabilities lay, and how they can be utilized by a threat. The threat can then lead to threat scenarios and unwanted incidents that affect the assets we want to protect. CORAS threat diagrams does not include if the system has the possibility to go back to a previous state, if a threat scenario or an unwanted incident occurs. The way CORAS threat diagrams are set up is not in the ideal manner for Markov chains, and it was a challenge in it self to separate it to able to fully use Markov chains.

If we applied Markov chains directly to CORAS threat diagram, there where two scenarios: The first was that the likelihood for the paths was not given and it was therefore impossible to calculate. The second scenario was that the likelihood given was linguistic and not numerical. The linguistic likelihood is again not useable if we wanted to apply Markov chains directly to CORAS threat diagram. If it was numerical likelihood we could apply Markov chains. The testing showed us that the only result we got was that we multiplied the given numerical likelihoods with each other. The use of probability matrix and applying the equations to calculate the probability only did this. The reason was that all of the other likelihoods that were essential to take full advantage of Markov chains where missing, and we had to assume that they were unknown. –Then why use Markov chains to calculate this simple equation? Neither did we fully use the full potential of Markov chains. We have to use a lot of time to make the probability matrix and insert the numbers in the equation, to just multiply the given likelihood in the end. The only solution to this was that is has to be made some changes before applying Markov chains. And if we wanted to make some changes then these changes must lead to full advantage of the potential Markov chains actually possess. These changes must also be in a matter that we preserve the essential in the CORAS threat diagrams, and at the same time introduce what is needed for Markov chains.

To find those paths that Markov chains can use we have to go back to the system, the brainstorming and historical data phase. We had to see how the system reacts when a threat that exploits those vulnerabilities in the system is introduced. We knew that CORAS threat diagram has a fixed path, therefore we would not look at the possibility that a state could

jump over a state and point to the next. Then it would be possible to lead to a next phase without exploiting a given vulnerability or a threat scenario. All these option are highly reflected in CORAS threat diagram. We decided that the paths and scenarios given in the CORAS threat diagram will not be questioned, because they are well evaluated and we would be testing something that is not decisive to our thesis.

We had to again look at the brainstorming and historical data phase, and simulate the system with respect to the CORAS threat diagram. We must simulate how the system will react when the scenarios given in the CORAS threat diagram happens. The result we are looking for is likelihoods for those paths that are needed for Markov chains. It is not for sure that all those paths we need are possible in the given system, and then we devote the likelihood of 0 for those paths when we insert them into Markov chains. The simulation gives also possibility to compare the result we get with the likelihood we have in the CORAS threat diagram. This can be seen as how effective the simulation went.

In our thesis we have used an algorithm when simulating. This algorithm takes our assumption and then calculates the likelihood. The algorithm does a random test to the given assumption. The main aspect in the simulation will always be the assumption. The assumption must be made through collecting all data of the system runs, the security mechanisms that are present in the system and how much they can help. If the assumptions made are not correct then we will not get correct answers. If we have some given likelihoods in the CORAS threat diagram then they can be compared to see how correct the simulated answer it is. If the given likelihood in the CORAS threat diagram is of linguistic value, then the simulations answer can be used to set up a numerical scale of the linguistic values.

We have tried to accomplish our success criteria's for this thesis when developing this method when relating Markov chains and CORAS threat diagrams. Our success criteria's were:

1. The thesis should compare Markov diagrams to CORAS diagrams with value to:

    - Expressiveness
    - Probability estimation
    - Tool-support
    - Purpose

We have to take look at how CORAS diagrams are built, especially CORAS threat diagrams. We know that CORAS diagrams are expressive with regards to that they are understandable, and show in diagrams where the threat can lay and what causes it can lead to. They are expressive also in the manner that they include an analyse team when making the diagrams, through brainstorming and historical data phase. CORAS diagrams probability estimation varies from how much information one gets from that phase, and if it is in numerical or linguistic value. This is the main area where it was thought that Markov chains could be a help.

Markov chains main expressiveness lays in its probability estimation ability. It has the ability to take different conditions a state can be under in its calculation. The relations between the states and their behaviour, and the way CORAS are built with their paths from a threat to an unwanted incident or a threat scenario is something that can resemble each other. We have seen that even if they resemble each other at some manner in a way, there are some changes that have to be made if we want to relate them to each other.

There are some tools available to calculate Markov chains, but there are not any tool support for relating CORAS diagrams and Markov chains. Because of some differences between both parts it is not an easy task to relate them before making some conclusive changes.

Besides proper analysis of exactly what, CORAS analysis wants to offer where the threat lays and how we can treat them, to show the probability for threats to occur. The purpose of Markov chains is to estimate the probability, and as we have said earlier this is where Markov chains can be a help for CORAS analysis.

2. The thesis should deliver a method integrating Markov analysis in risk analysis based on CORAS diagrams:

    - Makes it more reliable
    - Makes it more efficient
    - Gives more benefit on probability estimation

Our first interest was to apply Markov chains without making these changes. The likelihood is not always given in the CORAS threat diagram, but when it is given it can either be in linguistic or numerical value. When the likelihoods are not given or are in linguistic values then it is not possible to use Markov chains. Markov chains have its set of rules that also make us split the CORAS threat diagram. If the likelihood is given for all paths, and we apply Markov chains to calculate the probability we end up with only multiplying these values with each other. This is straightforward calculation of the probability and we do not need to introduce Markov chains to calculate this. It means that we have to go back to brainstorming and historical data phase and try to make a simulation of the system. There are rules that must be followed when simulating so we preserve both he basis of CORAS threat diagram, and what is needed to use Markov chains. The simulation must try to be done by looking at the paths and scenarios of the CORAS threat diagram.

We have tried to deliver a method of how Markov chains can be used in risk analysis based on CORAS diagrams. We have seen through our simulation and using the calculated values that it is possible to get reliable and competent calculations of the probability. This simulation lets us find the likelihoods we need for the use of Markov chains, and then use them for the calculation of the probability. The simulation has its basis in the CORAS threat diagram, so we have tried to preserve the idea of CORAS analysis in our simulation tool. We had to add new paths which were not part of CORAS threat diagram, but were essential for Markov chains. These new features shows what is needed to use the full potential of Markov chains.

Markov chains give the opportunity to calculate the probability in the way the system can behave when there is a threat, and this makes the answer more reliable. We also benefit because we can calculate the probability in different amount of steps of how a sequence can occur. Another advantage we get from introducing Markov chains is that we calculate the probability with much more expressiveness. We show that these are the sequences a threat can behave from one state to another state, depending on the amount of steps we choose to use.

Our simulation tool is to be looked as a beta program. It has the potential to be developed graphically and to be used on other threat diagrams as well. The time frame we have had on

this thesis, and the arduousness level of Markov chains and CORAS analysis semantics has not made it possible to make the simulation tool at its prospective. It was a great task and trial in it self to find the right course to how it is possible to relate these two.

In the aftermath we see that the simulation tool is something we could have developed better in terms of better usability and to be used on any threat diagram. It could use some further reconsideration on some issues. However, we have always had in mind that our goal is to find if and how it is possible to relate CORAS diagrams and Markov chains, and not a final implementation of some tool.

The program simulates in an understandable manner. We have ensured that all of the assumptions made are used when simulating the values, and that all of the other rules are followed as well. We have chosen to develop the simulation tool in active server page (ASP). The reason for this was our background in web-programming, and because of the time frame it was easier to develop the simulation tool using ASP. This simulation tool is designed to be a beta version that can be developed further. This simulation tool demonstrates how we can simulate the calculated values using the assumptions, and then be able to use them in Markov chains.

Our user manual resembles the simpleness of our simulation tool. It illustrates how to use the tool, and how to interpret the answers given from the simulation.

8. Conclusion

The motivation behind this thesis was how to make the probability estimation enhanced in CORAS analysis by relating CORAS diagrams and Markov chains. We limited our scope by looking at the differences and resembles between them, and how it is possible to relate them. We found that CORAS threat diagram is where the prospective is to introduce Markov chains.

The method we have developed captures the basis of CORAS threat diagram and introduces the Markov chains to it. We have based our research on capturing the features of Markov chains, and have tried to introduce this when we want to use Markov chains to calculate the probability in CORAS diagrams. The solution we saw for this is by simulating the system on the basis of the CORAS threat diagram. In the simulation we must present the new paths for each state that is needed to use Markov chain probability calculation. The simulation is based on assumptions made by looking at the historical data of the system, and going through brainstorming phase. These assumptions must be accurate, if not we end up with wrong answers.

The simulation tool that is made shows how the simulation is done, and how the answers could be used.

8.1 Future Work

From a narrow perspective, future work should consist of solidification of the work presented in this thesis. Investigating how to combine the simulation of the system with CORAS diagrams. Using CORAS diagrams to show where the threats lay, how they affect the system, and how it can be treated. And then use this as a basis when simulating the system and use the given values to calculate the probability with Markov chains.

Markov chains have the potential to enhance the probability estimation of CORAS diagram. We know that Markov chains need the absorbing and reverse likelihood in order to use its full potential. These paths likelihoods are not included in the CORAS diagrams. The other approach to look for future research is changing the matter CORAS threat diagrams are made by, and try to include the features needed for Markov chains in it. This will require changes in the way CORAS diagrams are made today.

It is also possible to look at other probability estimation methods that could be included, that do not need these changes. We know through this thesis that to calculate the probability using any other probability estimation methods we need the likelihood for each path. So the first phase should always be to give a numerical likelihood for each path in CORAS threat diagram.

Bibliography

[1] Thomas R. Peltier: “Information Security Risk Analysis”, Auerbach publications, 2001, ISBN 0-8493-0880-1

[2] An introduction to Stochastic Processes with Applications to Biology, by Linda J. S. Allen, Department of Mathematics and Statistics Texas Tech University.

[3] The Scandal of Father Brown, by Gilbert K. Chesterton

[4] http://coras.sourceforge.net/ (10.04.2007)

[5] The CORAS Model-based Method for Security Risk Analysis. By Folker den Braber, Gyrd Braendeland, Heidi E. I. Dahl, Iselin Engan, Ida Hogganvik, Mass S. Lund, Bjoernar Solhaug, Ketil Stoelen, Fredrik Vraalsen. SINTEF, Oslo. September 2006

[6] The Life and Work of A. A. Markov, by Gely P. Basharin[1], Amy N. Langville[2] and Valeriy A. Naumov[3] University, Moscow, Russia.[2]North Carolina State University, Raleigh, USA.[3]Lappeenranta University of Technology, Lappeenranta, Finland

[7] http://www.uio.no/studier/emner/matnat/ifi/INF5150/h06/ (12.03.07)

[8] Risk Analysis, System Analysis and Covey’s Seven Habits, Risk Analysis, Vol 21, No. 2, 2001. by Yacov Y.Haimes

[9] http://www.acusafe.com/Hazard_Analysis/HAZOP_Technique.pdf (05.07.2007)

[10] HAZOP, Hazard and Operability Study, by Marvin Rausand, Department of Production and Quality Engineering, NTNU, 07.10.2005(20.07.2007)

[11] Risiko Analyse, Veiledning til NS 5814, by Marvin Rausand, SINTEF 1991

[12] To Develope a HAZOP Study Teaching Module, by Choy, K K H, Hui, C W, Porter, J F and McKay, G, Department of Chemical Engineering, The Hong Kong Unversity of Science and Technology

[13] http://www.weibull.com (15.07.2007)

[14] http://fmeainfocentre.com/ (16.07.2007)

[15] Structured semantics for the CORAS security risk modelling lamguage by Heidi E.Dahl, Ida Hogganvik and Ketil Stoelen. SINTEF, Oslo. September 2007

[16] www.maplesoft.com (01.10.2007)

[17] Australian/New Zealand Standard AS/NZS 4360:2004
http://www.methodware.com/services/standards.shtml (20.09.2007)

[18] The SCORE method and tool, Master thesis by Stig Torsbakken,
University of Oslo, 2007

Appendix A

Discrete time Markov chains

[2]
Reducible / irreducible:
If there is only one *communication class*, then the Markov chain is irreducible, but if there is more than one communication class, then the Markov chain is reducible.

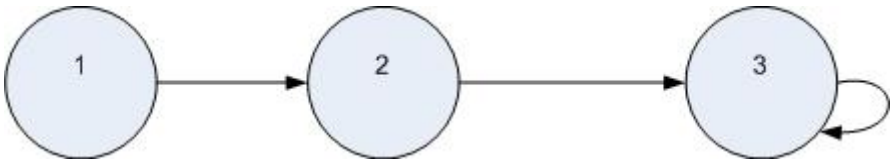
Communication classes:
The set of equivalence classes in a discrete time Markov chain are called the communication classes or classes of the Markov chain.
If every state in the Markov chain can be reached from every other state, then there is only one communication class (all the states are in the same class)

Periodic / aperodic:
The period of state i, denoted as $d(i)$, is the greatest common divisor of all integers $n \geq 1$ for which $p^{(n)}_{ii} > 0$; that is:

$$d(i) = g.c.d\{n | p^{(n)}_{ii} > 0 \text{ and } n \geq 1\}$$

If a state i has a period of $d(i) > 1$, it is said to be periodic of period $d(i)$. If the period of a state equals one, it is said to be aperiodic.

Example:



The corresponding transition matrix for this graph is:

$$P = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

It is easy to see that there are three communication classes, {1}, {2}, and {3}. The value of $d(i) = 0$ for $i = 1, 2$ because $p^{(n)}_{ii} = 0$ for $i = 1, 2$ and $n = 1, 2, \ldots$. Also, $d(3) = 1$; state 3 is aperiodic.

Recurrent / transient:

State i is transient if $\sum_{n=1}^{\infty} f_{ii}^{(n)} < 1$. State i is recurrent if $\sum_{n=1}^{\infty} f_{ii}^{(n)} = 1$.

If state i is recurrent, then the set $\{f_{ii}^{(n)}\}_{n=0}^{\infty}$ defines a probability distribution for the random variable representing the first return time. When state i is transient, $\{f_{ii}^{(n)}\}_{n=0}^{\infty}$, does not define a complete set of probabilities necessary to define a probability distribution.

Continuous time Markov chains

[2]

Reducible / irreducible:

Same definition as in discrete. If every state can be reached from every other state, it is irreducible; otherwise it is reducible.

Periodic / aperodic:

There is no concept of periodic and aperiodic in continuous time Markov chains because the intervened time is random.

Recurrent / transient:

The definition is the same as in discrete time Markov chains.

State i is recurrent (transient) in a continuous time Markov chain $\{X(t)\}, t \geq 0$, if the first return time is finite (infinite),

$$\text{Prob}\{T_{ii} < \infty | X(0) = 1\} = 1 \ (< 1)$$

Appendix B

```
<p><img border="0" src="images/simulering_1.gif"></p>
<p> </p>

<table border="0" width="48%" cellspacing="0" cellpadding="0">
<tr>
<td>The parameters</td>
</tr>
<tr>
<td>
<form method="POST" action="sim1.asp">
<table border="0" width="100%" cellspacing="0" cellpadding="0">
<tr>
        <td width="3%"> </td>
        <td width="20%"> </td>
        <td width="13%"> </td>
        <td width="1%"> </td>
        <td width="27%">Simulation period</td>
        <td width="35%">
        <input type="text" name="days" size="20" value="20">
        years</td>
</tr>
<tr>
        <td width="3%"> </td>
        <td colspan="5"><br>
        Number of accurrences</td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="20%">Phase A-C ( a )</td>
        <td width="13%">
        <input type="text" name="attacks_ac" size="7"
value="<%=request.form("attacks_ac")%>"></td>
        <td width="1%"> </td>
        <td width="27%">Phase C-A ( c )</td>
        <td width="35%">
        <input type="text" name="attacks_ca" size="7"
value="<%=request.form("attacks_ca")%>"></td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="20%">Phace C-F ( d )</td>
        <td width="13%">
        <input type="text" name="attacks_cf" size="7"
value="<%=request.form("attacks_cf")%>"></td>
        <td width="1%"> </td>
        <td width="27%">Phase F-C ( f )</td>
        <td width="35%">
```

```
        <input type="text" name="attacks_fc" size="7"
value="<%=request.form("attacks_fc")%>"></td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="20%"> </td>
        <td width="13%"> </td>
        <td width="1%"> </td>
        <td width="27%">Phase F-A ( g ) </td>
        <td width="35%">
        <input type="text" name="attacks_fa" size="7"
value="<%=request.form("attacks_fa")%>"></td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="20%">Phase F-H ( h ) </td>
        <td width="13%">
        <input type="text" name="attacks_fh" size="7"
value="<%=request.form("attacks_fh")%>"></td>
        <td width="1%"> </td>
        <td width="27%">Phase H-F ( j ) </td>
        <td width="35%">
        <input type="text" name="attacks_hf" size="7"
value="<%=request.form("attacks_hf")%>"></td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="20%"> </td>
        <td width="13%"> </td>
        <td width="1%"> </td>
        <td width="27%">Phase H-C ( k )</td>
        <td width="35%">
        <input type="text" name="attacks_hc" size="7"
value="<%=request.form("attacks_hc")%>"></td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="20%"> </td>
        <td width="13%"> </td>
        <td width="1%"> </td>
        <td width="27%">Phase H-A ( l )</td>
        <td width="35%">
        <input type="text" name="attacks_ha" size="7"
value="<%=request.form("attacks_ha")%>"></td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="20%"> </td>
        <td width="13%"> </td>
        <td width="1%"> </td>
```

```
        <td width="27%"> </td>
        <td width="35%">
         </td>
</tr>
<tr>
        <td width="3%"> </td>
        <td width="96%" colspan="5">
        <font color="#FF0000">* Rules from the thesis<br>
        c=&lt;a, d=&lt;a &amp; f=&lt; d, h=&lt;d, j=&lt;h, k=&lt;h, l=&lt;h,
g=&lt;d;f+g&lt;=d,
        j+k+l&lt;h</font></td>
</tr>
</table>
<p><input type="submit" value="Simulate" name="B1"></p>
</form>
</td>
</tr>
</table>
<%

'get variables

error = false

dager=Request.Form("days")
angrep=Request.Form("attacks")

aValue = cInt(request.form("attacks_ac"))
cValue = cInt(request.form("attacks_ca"))
dValue = cInt(request.form("attacks_cf"))
fValue = cInt(request.form("attacks_fc"))
gValue = cInt(request.form("attacks_fa"))
hValue = cInt(request.form("attacks_fh"))
jValue = cInt(request.form("attacks_hf"))
kValue = cInt(request.form("attacks_hc"))
lValue = cInt(request.form("attacks_ha"))


'validate number of days
If (dager<>"" and dager>0) Then
'rule checks

'Regler  c=<a, d=<a & f=< d, h=<d, j=<h, k=<h, l=<h, g=<d;f+g<=d, j+k+l<h

errorstring =""
if(cValue>aValue) then
error= true
errorstring = errorstring & "The assumption for c is greater than the assumption for a <br>"
end if
```

```
if(dValue>aValue) then
error =true
errorstring = errorstring & "The assumption for d is greater than the assumption for a <br>"
end if

if(fValue>dValue) then
error = true
errorstring = errorstring & "The assumption for f is greater than the assumption for d <br>"
end if

if(hValue>dValue) then
error = true
errorstring = errorstring & "The assumption for h is greater than the assumption for d <br>"
end if

if(jValue>hValue) then
error = true
errorstring = errorstring & "The assumption for j is greater than the assumption for h <br>"
end if

if(kValue>hValue) then
error = true
errorstring = errorstring & "The assumption for k is greater than the assumption for h <br>"
end if

if(lValue>hValue) then
error = true
errorstring = errorstring & "The assumption for l is greater than the assumption for h <br>"
end if

if (gValue>dValue) then
error = true
errorstring = errorstring & "The assumption for g is greater than the assumption for d<br>"
end if

if((fValue + gValue)>dValue) then
error = true
errorstring = errorstring & "The sum of the assumption for f and g is greater than the
assumption for d<br>"
end if

if((jValue + kValue + lValue)>hValue) then
error = true
errorstring = errorstring & "The sum of the assumption for j,k and l is greater than the
assumption for h<br>"

end if


if(bValue<0) then
```

```
error = true
errorstring = errorstring & "The assumption for c cannot be a negative value<br>"
end if

if(eValue<0) then
error = true
errorstring = errorstring & "The assumption for e cannot be a negative value<br>"
end if

if(iValue<0) then
error = true
errorstring = errorstring & "The assumption for i cannot be a negative value<br>"
end if

if(mValue<0) then
error = true
errorstring = errorstring & "The assumption for m cannot be a negative value<br>"
end if



        if(error=false) then

        'Get assumptions
        fa = (RandomNumber(aValue,aValue,dager,"a"))
        fc = (RandomNumber(cValue,aValue,dager,"c"))
        fd = (RandomNumber(dValue,aValue,dager,"d"))
        ff = (RandomNumber(fValue,dValue,dager,"f"))
        fh = (RandomNumber(hValue,dValue,dager,"h"))
        fj = (RandomNumber(jValue,hValue,dager,"j"))
        fk = (RandomNumber(kValue,hValue,dager,"k"))
        fg = (RandomNumber(gValue,dValue,dager,"g"))
        fl = (RandomNumber(lValue,hValue,dager,"l"))

        response.write(fa & fc & fd & ff & fh & fj & fk & fg & fl)

        response.write("<br>")




        aValue = cDbl(getNumber(fa))
        bValue = 1- cDbl(getNumber(fa))
        cValue = cDbl(getNumber(fc))
        dValue = cDbl(getNumber(fd))
        eValue = 1 - cDbl(getNumber(fd)) - cDbl(getnumber(fc))
        fValue = cDbl(getNumber(ff))
        gValue = cDbl(getNumber(fg))
        hValue = cDbl(getNumber(fh))
```

```
        iValue = 1 - cLng(getnumber(fh))- cDbl(getnumber(ff)) - cDbl(getNumber(fg))
        jValue = cDbl(getNumber(fj))
        kValue = cDbl(getNumber(fk))
        lValue = cDbl(getNumber(fl))
        mValue = 1 - cLng(getNumber(fj)) - cDbl(getNumber(fk))- cDbl(getNumber(fl))

        response.write("b : " & bValue & "<br>")
        response.write("e : " & eValue & "<br>")
        response.write("i ; " & iValue & "<br>")
        response.write("m : " & mValue & "<br>")

        if(bValue<0) then
                response.write("<br>b less then 0<br>")
        end if

        if(eValue<0) then
                response.write("<br>e less then 0<br>")
        end if

        if(iValue<0) then
                response.write("<br>i less then 0<br>")
        end if

        if(mValue<0) then
                response.write("<br>m less then 0<br>")
        end if


        response.write("<p><h4>Calculating the probability from State A -> State H, using
Markov chains:</h4>")

        response.write("In 3-steps: ")
        response.write(aValue * dValue * hValue)
        response.write("<br>In 5 steps: ")
        response.write(((bValue*bValue + aValue*cValue)*aValue*dValue +
(bValue*aValue + aValue*eValue)* (eValue*dValue + dValue*iValue) +
aValue*dValue*(dValue*f + iValue*iValue + hValue*jValue))*hValue + ((bValue*aValue +
aValue*eValue)*dValue*hValue + aValue*dValue*(iValue*hValue +
hValue*mValue))*mValue)
        response.write("<br>In 6 steps: ")
        response.write(((bValue*bValue + aValue*cValue)*bValue +(bValue*aValue +
aValue*eValue)*cValue + aValue*dValue*gValue)*aValue*dValue*hValue +
((bValue*bValue + aValue*cValue)*aValue + (bValue*aValue + aValue*eValue)*eValue +
aValue*dValue*fValue)*((eValue*dValue + dValue*iValue)*hValue  +
dValue*hValue*mValue) + ((bValue*aValue + aValue*eValue)*dValue +
aValue*dValue*iValue)*((dValue*fValue +iValue*iValue + hValue*jValue)*hValue +
(iValue*hValue + hValue*mValue)*mValue) + aValue*dValue*hValue*((k*dValue +
jValue*iValue + mValue*jValue)*hValue + (hValue*jValue + mValue*mValue) *mValue))
```

```
        else

        response.write("<h4>The simulation can not be executed. Please check your
        assumtions:</h4>"+errorstring)


        end if


End If

Function getNumber(strValue)
getNumber = replace(trim(mid(strValue,instr(strValue,":")+1,len(strValue)-4)),"<hr>","")
End Function



Function RandomNumber(intHighestNumber, ofValue,loops, sValue)

valueTable =""
startTable
="<table><tr><td><table><tr><td><strong>Attacks</strong></td></tr><tr><td><strong>Da
y</strong></td></tr></table></td>"

for i=1 to dager
valueTable = valueTable & "<td><table width=""20"">"
Randomize()
rndValue= int(rnd() * (intHighestNumber+1))

valuetable = valuetable & "<tr><td>" & rndValue & "/" & ofValue & "</td></tr>"
valuetable = valuetable & "<tr><td>" & i & "</td></tr>"
valueSum= valueSum+ (rndValue/ofValue)

valueTable = valueTable & "</table></td>"
next


endTable = "</tr></table>"

RandomNumber=starttable & valueTable & endTable & " " & sValue & " value : " &
valueSum/loops & "<hr>"

End Function

%>
```

User manual

This user manual guides the user through the steps which are required to perform a simulation to determine the likelihood values, after the assumption values have been inserted for the particular phases in a state diagram. This process will be done through running a web application running which can be accessed on the URL:

http://www.semat.no/simulering/

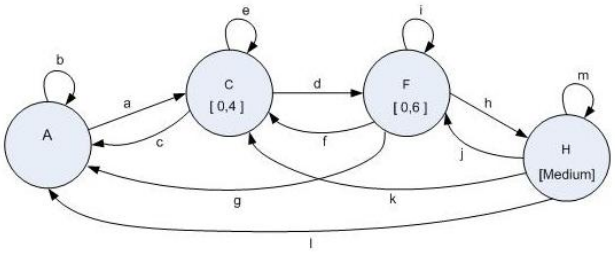
Following describes the user steps:

1. **Running the application**
   Open a web browser, i.e. Internet Explorer or Mozilla Firefox and enter the URL:
   http://www.semat.no/simulering/

2. **Choosing a state diagram**
   A page with the particular state diagram will be listed (figure 1), please select det state diagram you want to use to fulfill the simulation

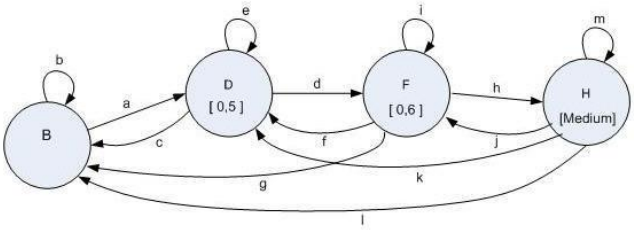
Choose a substate diagram

Sub state diagram 1



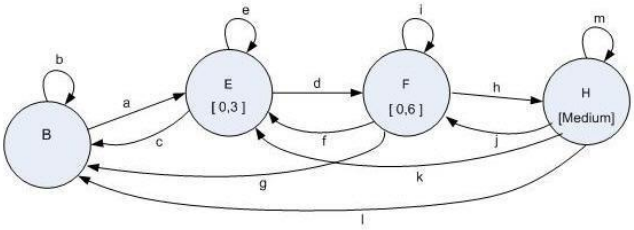
Simulate

Sub state diagram 2



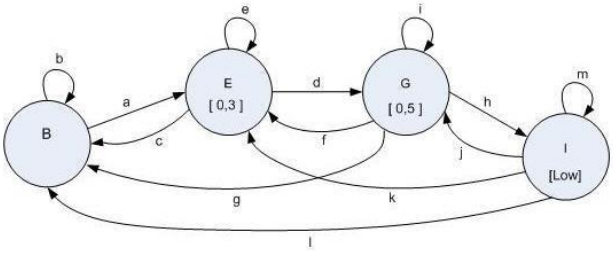
Simulate

Sub state diagram 3



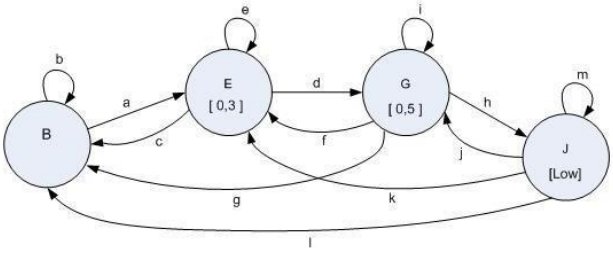
Simulate

Sub state diagram 4



Simulate

Sub state diagram 5



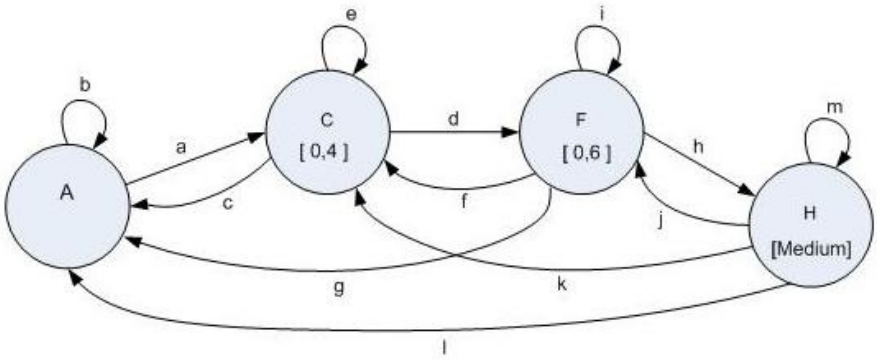
simulate

Figure 1: Selecting a sub state diagram

3. Fill inn assumption values

When a state diagram has been selected, a form will be displayed on the next page(figure 2) .

Please fill inn the assumption fields according to the rules from the thesis. These rules are will be displayed marked with red under the assumption fields. If invalid values are entered, the errors will be prompted to the user.



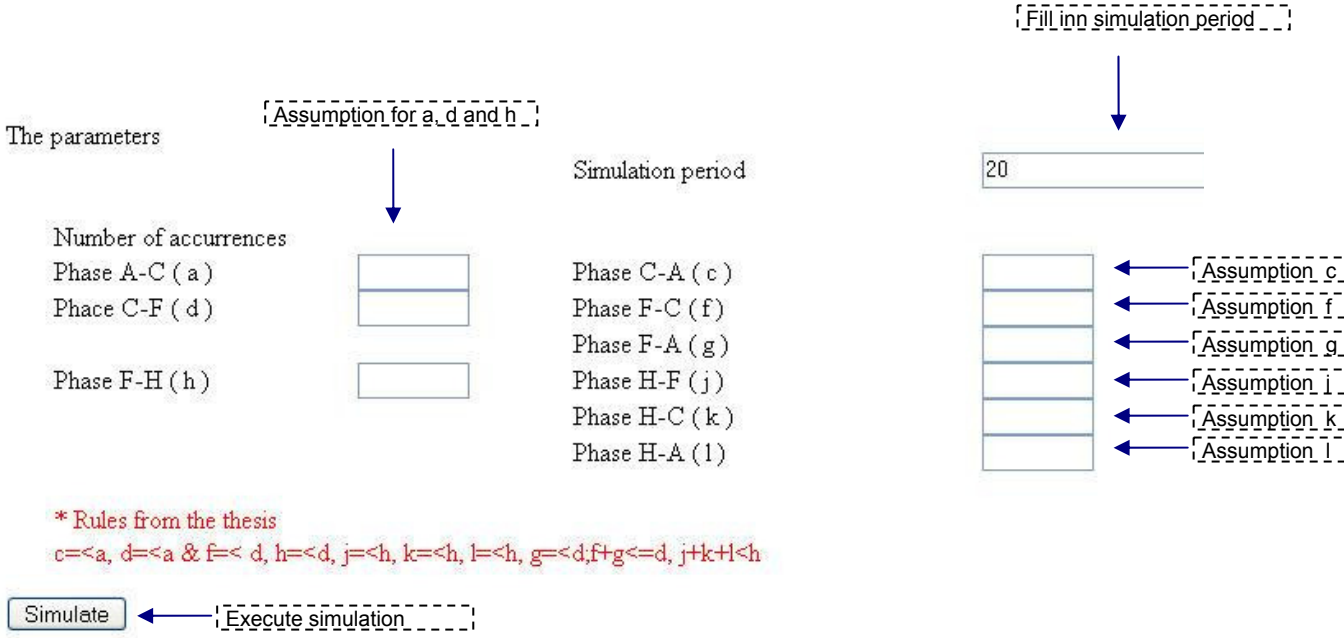
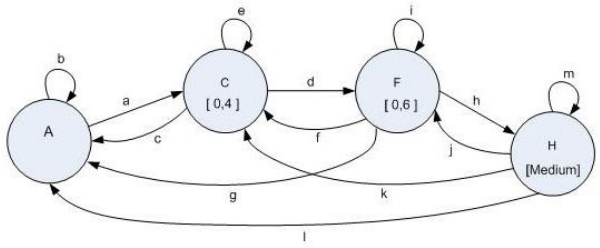


Figure 1 User manual: Fill inn assumption values

4. Displaying the result

When pressed the simulate button, the values are been passed to the application and the likelihoods are calculated and the result values are shown as the illustrated on the next page.



The parameters

Simulation period: 20 years

Number of accurrences			
Phase A-C (a)	10	Phase C-A (c)	5
Phace C-F (d)	8	Phase F-C (f)	2
		Phase F-A (g)	2
Phase F-H (h)	6	Phase H-F (j)	2
		Phase H-C (k)	1
		Phase H-A (1)	1

* Rules from the thesis
c=<a, d=<a & f=< d, h=<d, j=<h, k=<h, l=<h, g=<d;f+g<=d, j+k+l<h

Simulate

Attacks	9/10	8/10	5/10	10/10	8/10	6/10	9/10	0/10	7/10	7/10	4/10	2/10	7/10	1/10	4/10	7/10	3/10	5/10	2/10	3/10
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

a value : 0,534999990463257 ← Likelihood for phase a

Attacks	4/10	3/10	3/10	2/10	4/10	4/10	0/10	3/10	0/10	0/10	0/10	1/10	2/10	3/10	0/10	0/10	1/10	2/10	2/10	5/10
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

c value : 0,194999992847443 ← Likelihood for phase c

Attacks	2/10	5/10	8/10	3/10	2/10	0/10	5/10	7/10	7/10	7/10	8/10	5/10	4/10	5/10	1/10	4/10	3/10	6/10	7/10	7/10
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

d value : 0,480000019073486 ← Likelihood for phase d

Attacks	2/8	0/8	1/8	1/8	0/8	2/8	2/8	0/8	2/8	2/8	0/8	0/8	1/8	2/8	1/8	1/8	2/8	0/8	0/8	2/8
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

f value : 0,13125 ← Likelihood for phase f

Attacks	3/8	5/8	3/8	4/8	0/8	4/8	5/8	5/8	0/8	0/8	3/8	6/8	2/8	1/8	0/8	2/8	4/8	6/8	2/8	1/8
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

h value : 0,35 ← Likelihood for phase h

Attacks	0/6	2/6	2/6	0/6	0/6	2/6	0/6	1/6	1/6	1/6	2/6	0/6	0/6	0/6	0/6	1/6	2/6	2/6	1/6	2/6
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

j value : 0,158333325386047 ← Likelihood for phase j

Attacks	1/6	0/6	0/6	0/6	0/6	1/6	1/6	1/6	0/6	1/6	0/6	0/6	1/6	1/6	0/6	0/6	1/6	0/6	1/6	1/6
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

k value : 8,33333253860474E-02 ← Likelihood for phase k

Attacks	0/8	0/8	1/8	0/8	1/8	0/8	1/8	2/8	0/8	2/8	0/8	2/8	0/8	2/8	1/8	0/8	0/8	1/8	0/8	1/8
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

g value : 0,0875 ← Likelihood for phase g

Attacks	1/6	0/6	0/6	0/6	1/6	1/6	1/6	0/6	0/6	0/6	0/6	1/6	1/6	1/6	0/6	0/6	0/6	0/6	0/6	1/6
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

l value : 6,66666626930237E-02 ← Likelihood for phase l

b : 0,465000009536743 ← Likelihood for phase b, e, i, m
e : 0,324999988079071
i ; 0,78125
m : 0,850000011920929

Calculating the probability from State A -> State H, using Markov chains: ← The total probability, using Markov chains

In 3-steps: 8,98800019693373E-02
In 5 steps: 0,3521784524294
In 6 steps: 0,501698526148471